ESE 531: Digital Signal Processing

Lec 23: April 19, 2018

Wavelet Transform, Compressive Sensing

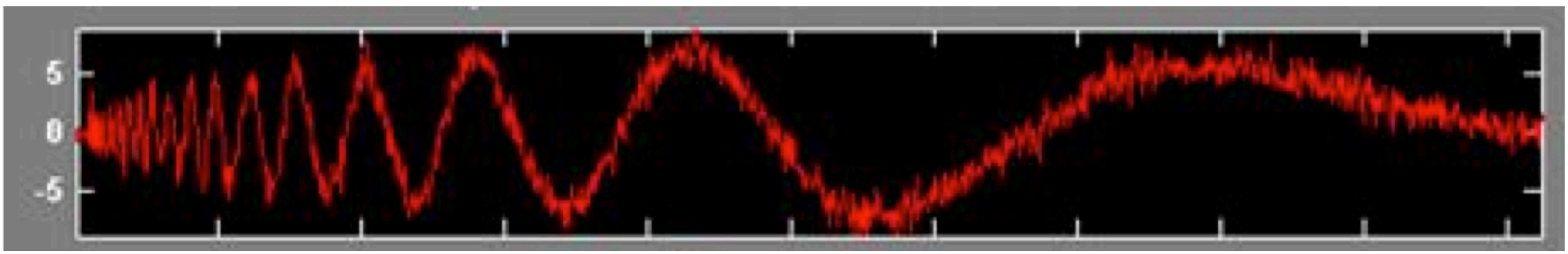
Previously

- Today
  - Wavelet Transform
  - Compressive Sampling/Sensing

Wavelet Transform

Motivation

- Some signals obviously have spectral characteristics that vary with time



Criticism of Fourier Spectrum

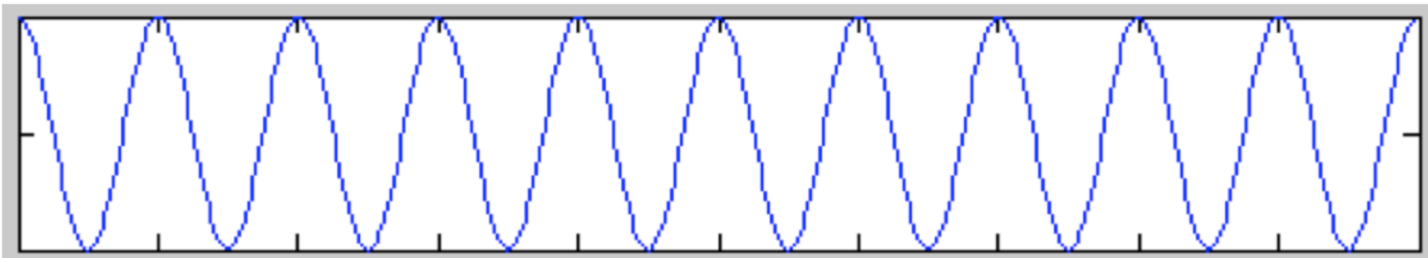
- It's giving you the spectrum of the 'whole time-series'

- Which is OK if the time-series is stationary. But what if its not?

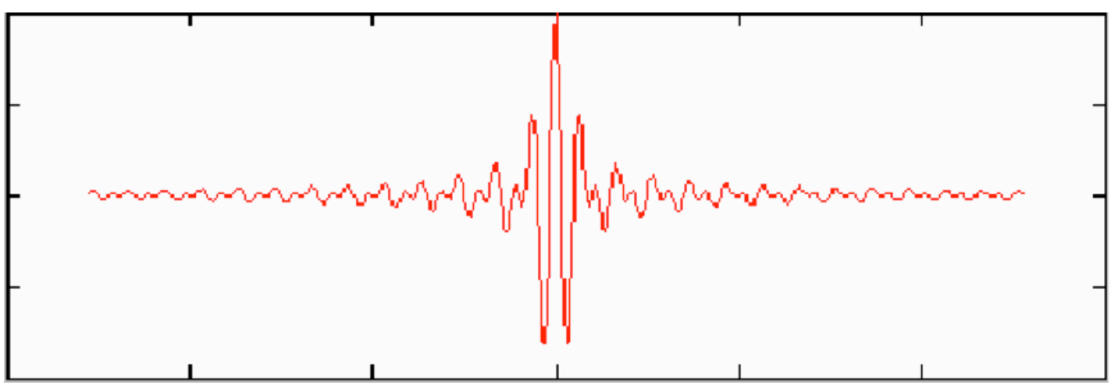
- We need a technique that can "march along" a time series and that is capable of:
  - Analyzing spectral content in different places
  - Detecting sharp changes in spectral character

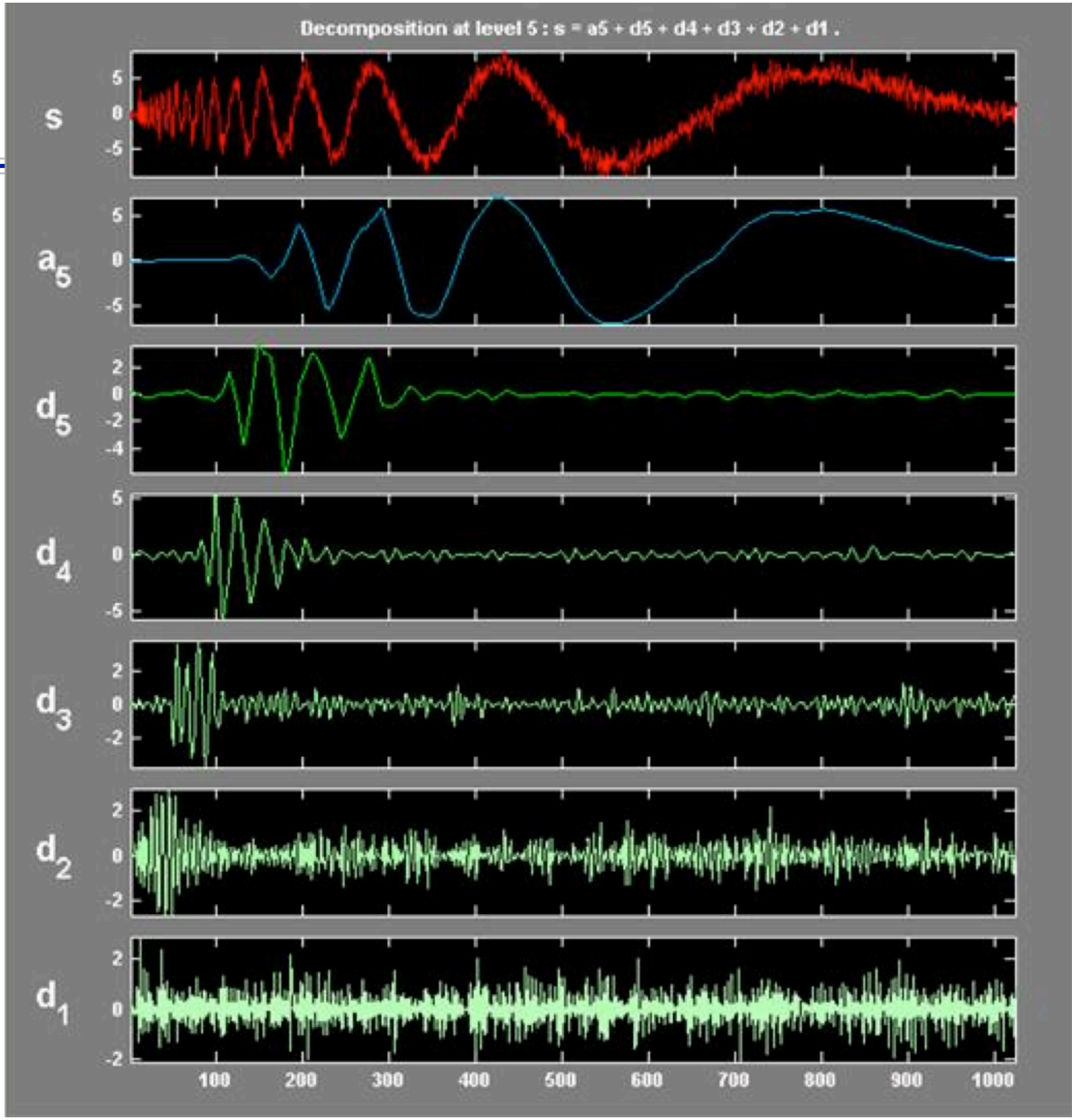
Fourier vs. Wavelet

- Fourier Analysis is based on an indefinitely long cosine wave of a specific frequency



- Wavelet Analysis is based on an short duration wavelet of a specific center frequency




Decomposition at level 5 : s = a5 + d5 + d4 + d3 + d2 + d1 .
s
a5
d5
d4
d3
d2
d1
100
200
300
400
500
600
700
800
900
1000

Wavelet Transform

- All wavelet derived from *mother* wavelet

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right)$$

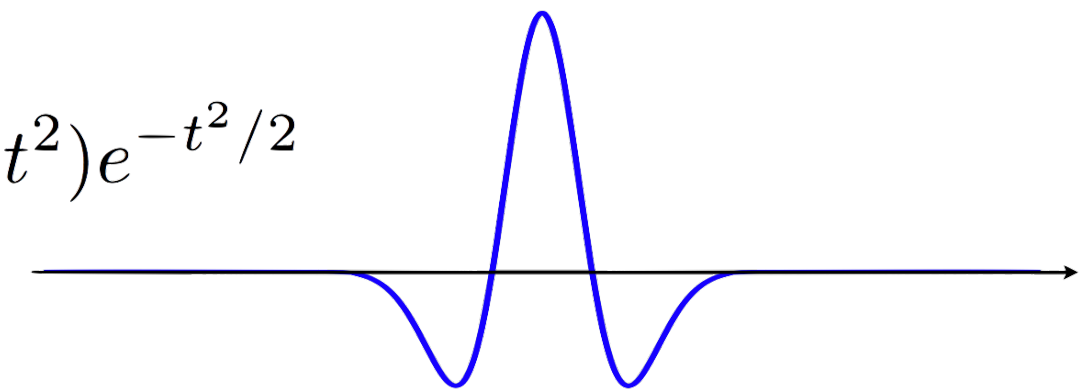
- Must satisfy

$$\int_{-\infty}^{\infty} |\Psi(t)|^2 dt = 1 \quad \Rightarrow \text{unit norm}$$

Examples of Wavelets

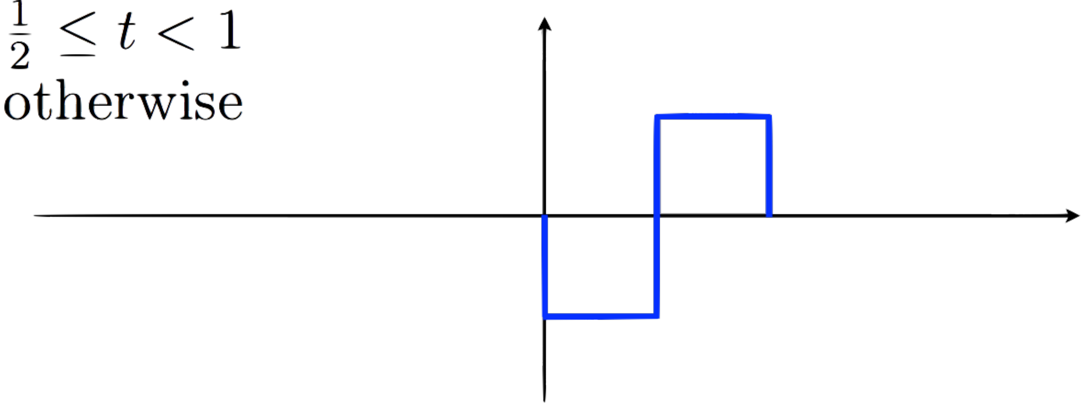
Mexican Hat

$$\Psi(t) = (1 - t^2)e^{-t^2/2}$$



Haar

$$\Psi(t) = \begin{cases} -1 & 0 \leq t < \frac{1}{2} \\ 1 & \frac{1}{2} \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$$



Wavelet – Scaled and Shifted

normalization

shift in time

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\,\psi\!\left(\frac{t-\tau}{s}\right)$$

change in scale:
big s means long
wavelength

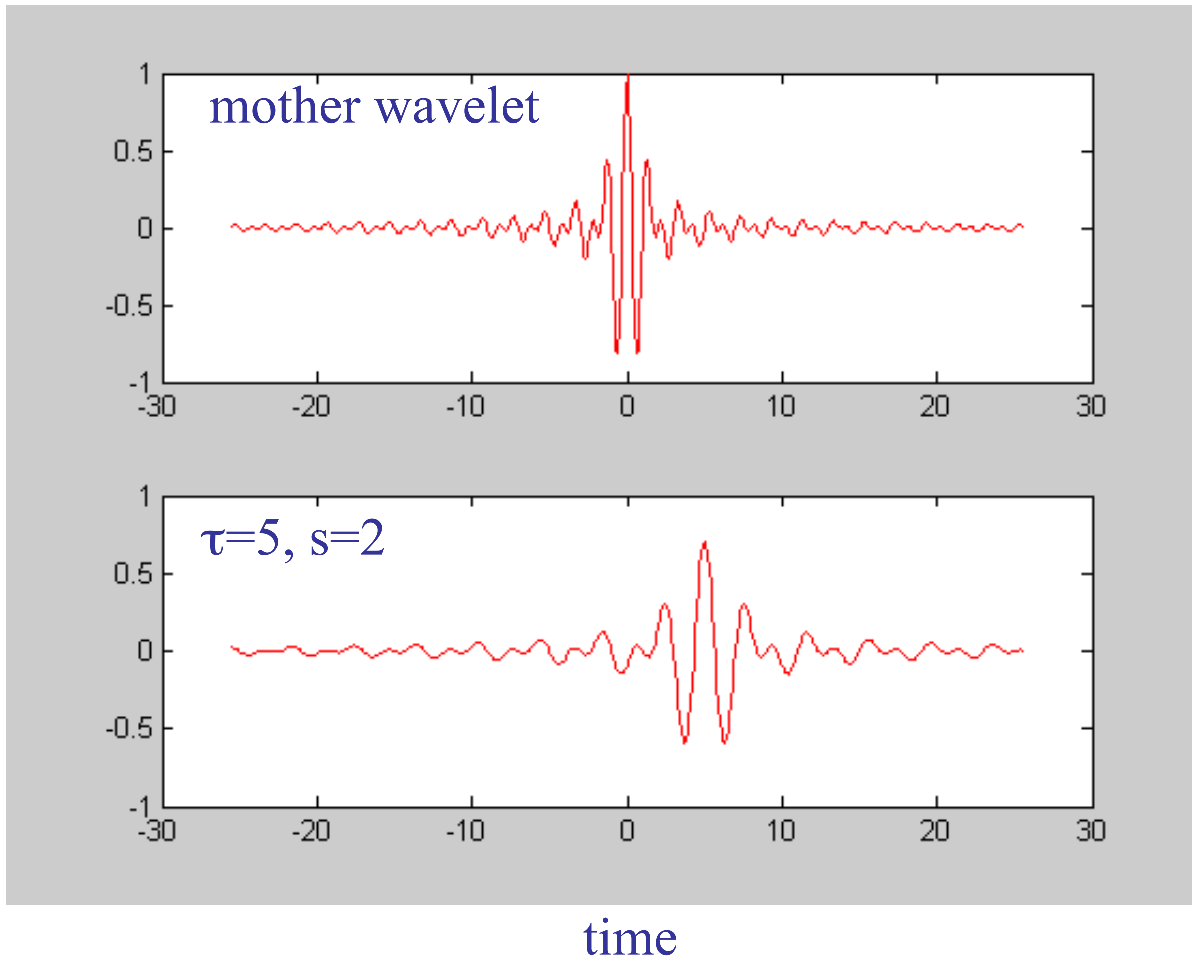
wavelet with
scale, s and time, τ

Mother wavelet

Shannon Wavelet

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t-\tau}{s}\right)$$

- Ψ(t) = 2 sinc(2t) – sinc(t)



Wavelet Transform

time-series

$$\gamma(s,t) = \int f(t)\Psi_{s,\tau}(t)dt$$

coefficient of wavelet
with
scale, s and time, τ

wavelet with
scale, s, and shift, τ

Inverse Wavelet Transform

- Build up a time-series as sum of wavelets of different scales, s, and positions, t

$$f(t) = \int\int \gamma(s, \tau)\psi_{s,\tau}(t)\,d\tau\,ds$$

time-series

coefficients of wavelets

wavelet with scale, s and time, τ

Wavelet Transform

- Determining the wavelet coefficients for a fixed scale, *s*, can be thought of as a filtering operation

$$\gamma(s,t) = \int f(t)\Psi_{s,\tau}(t)dt$$

$$\gamma_s(t) = \int f(t)\Psi_s(t)\,dt = f(t) * \Psi_s(t)$$

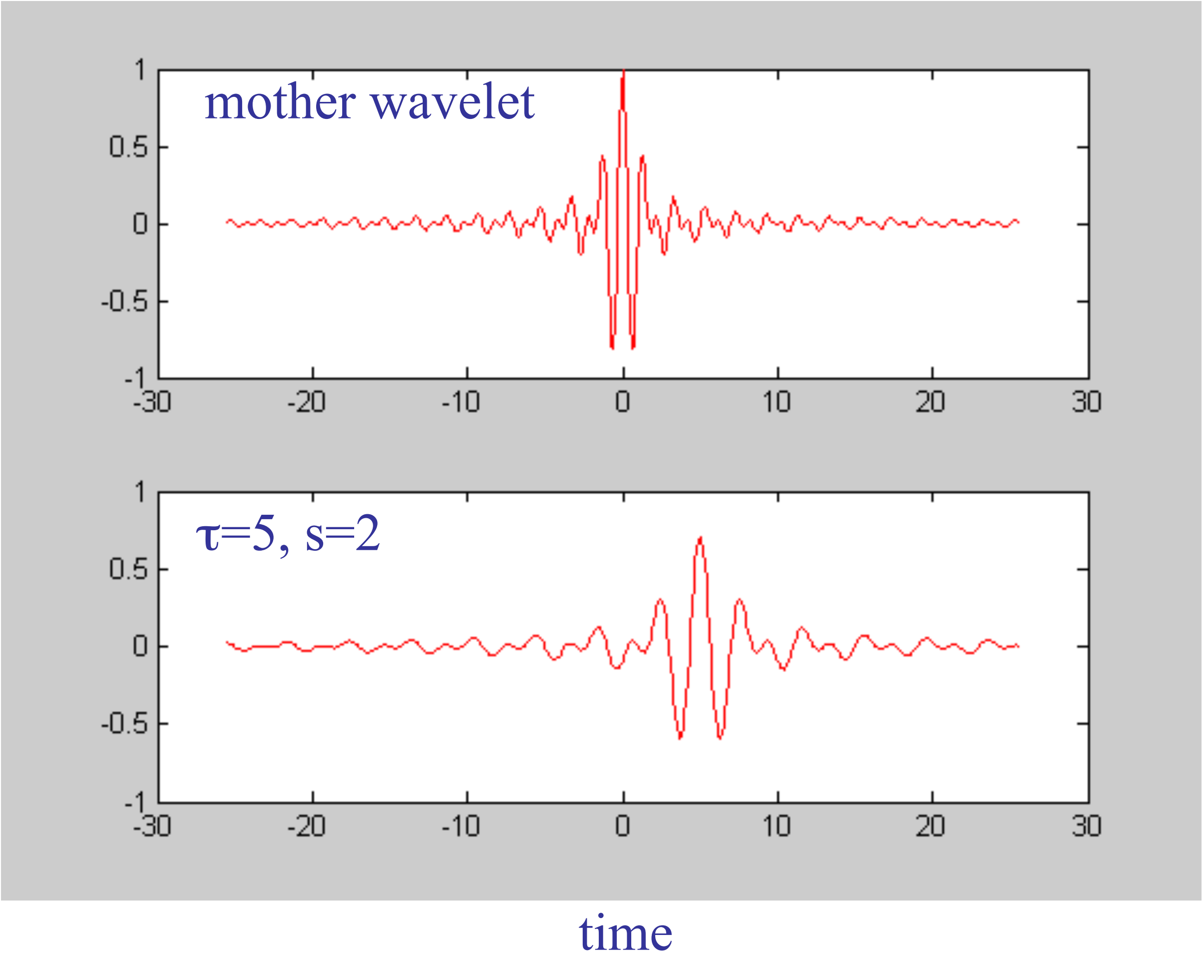
- where

$$\Psi_s(t) = \frac{1}{\sqrt{s}}\Psi(\frac{t}{s})$$

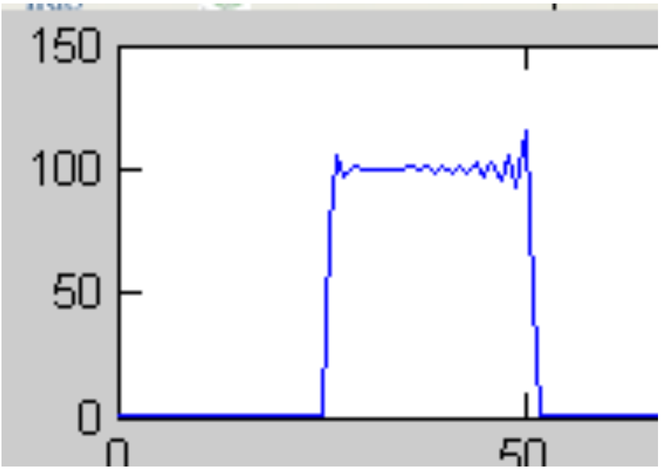
Shannon Wavelet

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t-\tau}{s}\right)$$

- $\Psi(t) = 2\,\mathrm{sinc}(2t) - \mathrm{sinc}(t)$



Fourier spectrum of Shannon Wavelet



frequency, ω

- Wavelet coefficients are a result of bandpass filtering

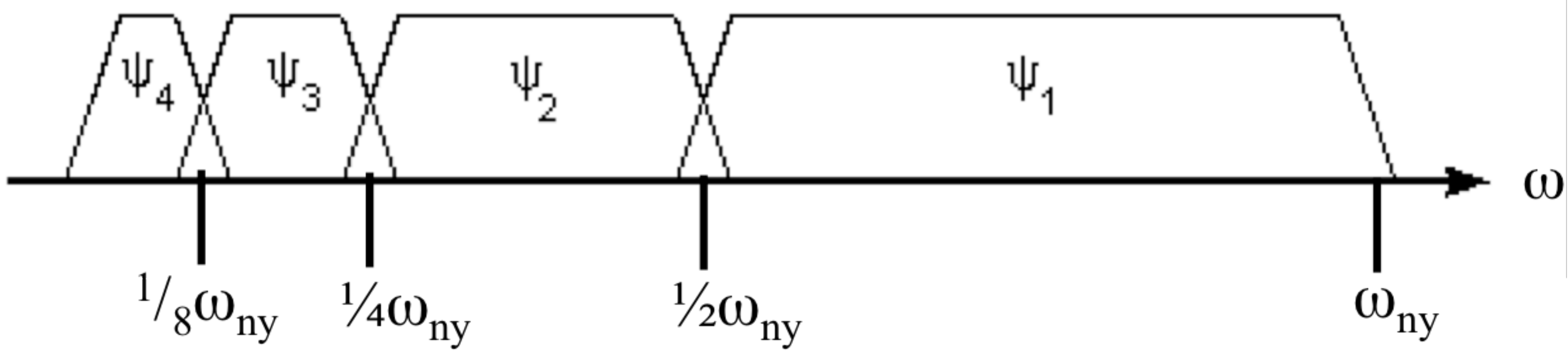
Discrete wavelets:

- Scale wavelets only by powers of 2
  - $s_j=2^j$
- And shifting by multiples of s_j for each successive scale
  - $\tau_{j,k} = 2^j k$
- Then $\gamma(s_j, \tau_{j,k}) = \gamma_{jk}$
  - where $j = 1, 2, \ldots\infty$, $k = -\infty\ldots -2, -1, 0, 1, 2, \ldots\infty$

$$\gamma_{j,k} = \frac{1}{\sqrt{2^j}} \int f(t)\Psi\left(\frac{t-k2^j}{2^j}\right)dt$$

Discrete Wavelet Transform

- The factor of two scaling means that the spectra of the wavelets divide up the frequency scale into octaves (frequency doubling intervals)

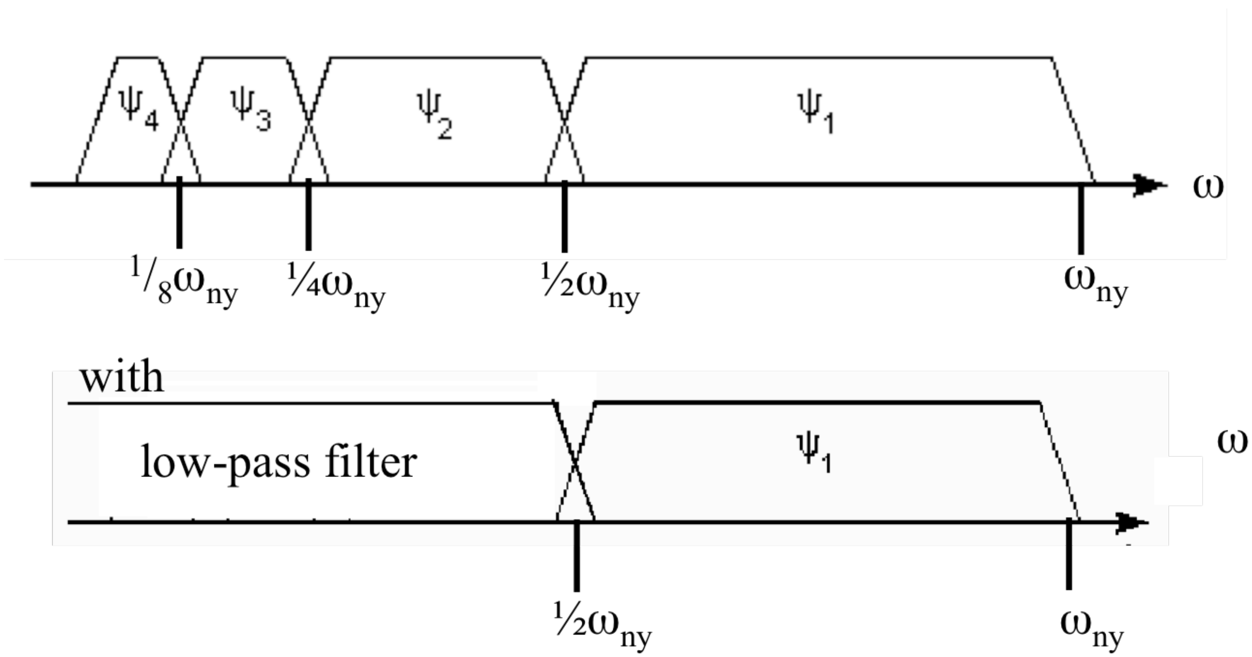


Discrete Wavelet Transform

- As we saw previously, the coefficients of Ψ is just the band-pass filtered time-series, where Ψ is the wavelet, now viewed as the impulse response of a bandpass filter.

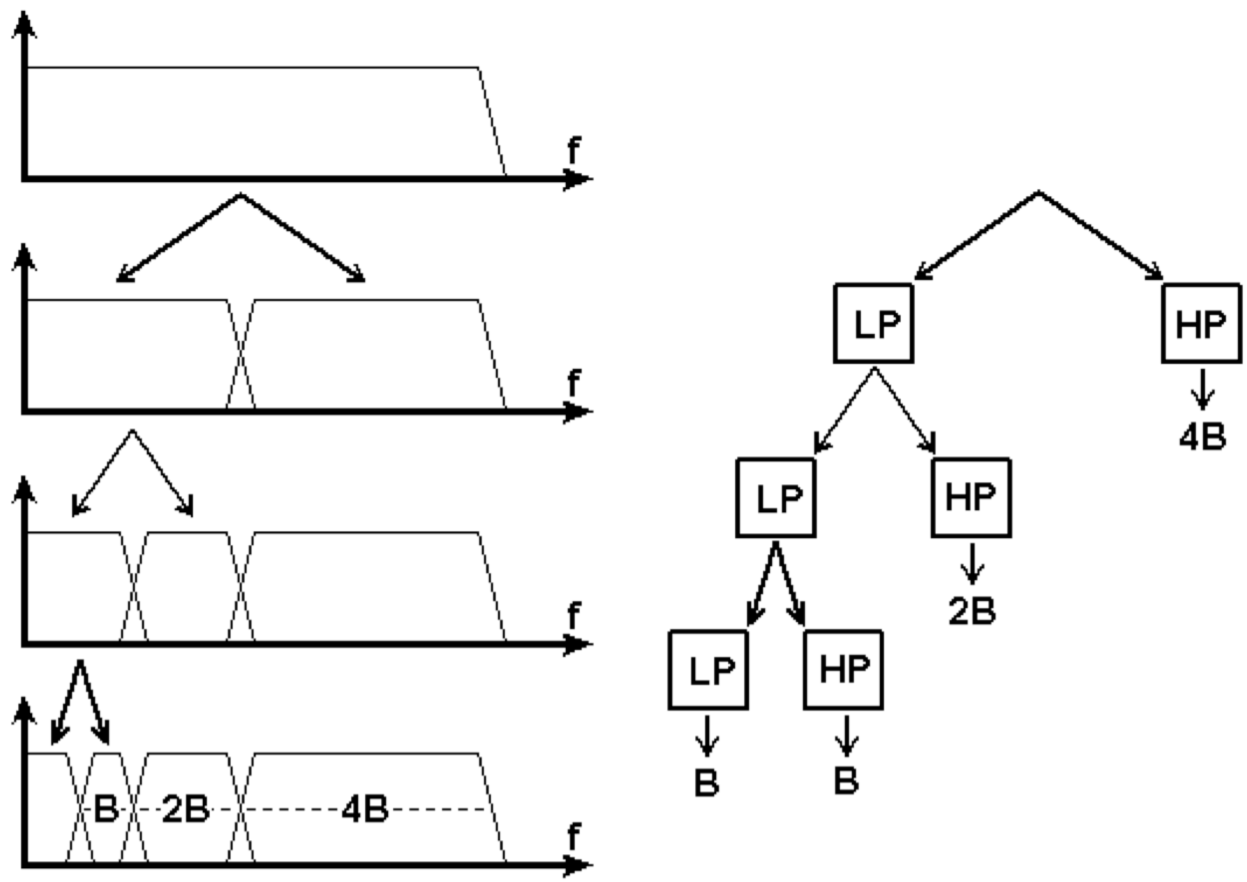
Discrete Wavelet Transform

- As we saw previously, the coefficients of Ψ is just the band-pass filtered time-series, where Ψ is the wavelet, now viewed as the impulse response of a bandpass filter.

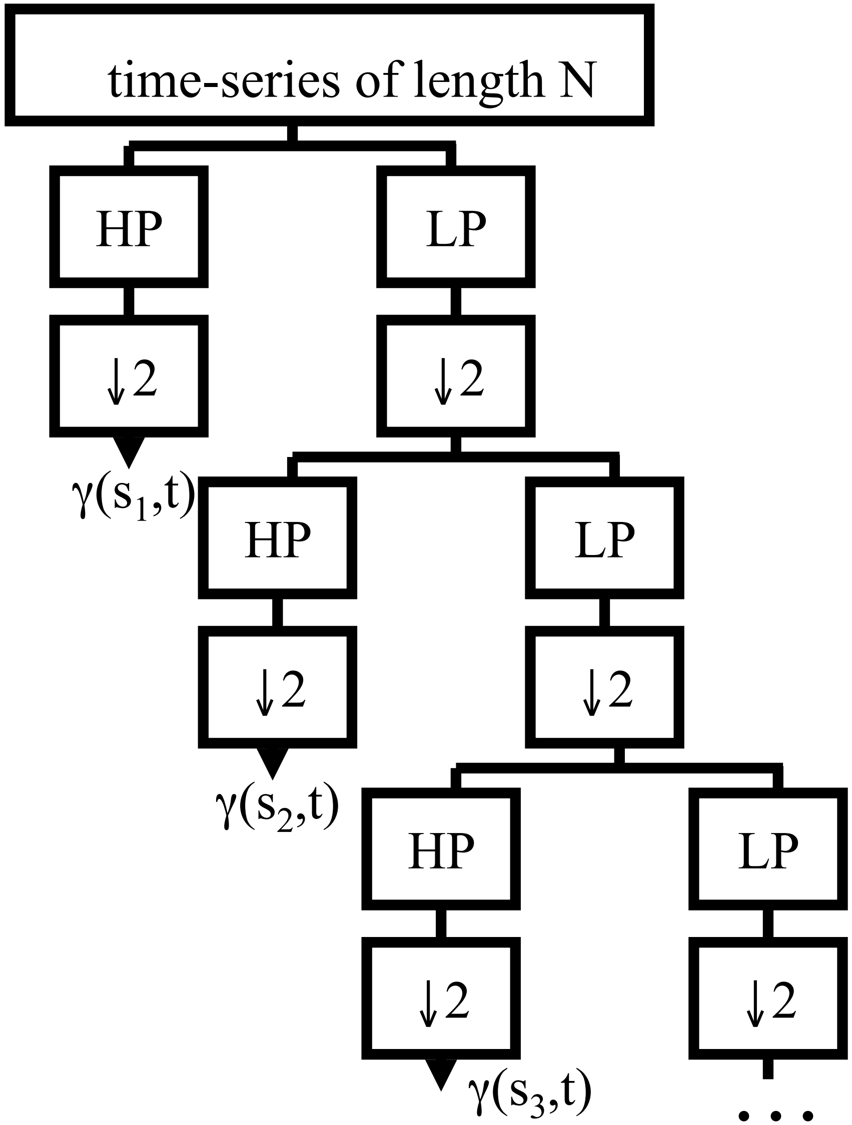


Digital Wavelet as Multirate Filter Bank

- Repeat recursively!



Digital Wavelet as Multirate Filter Bank

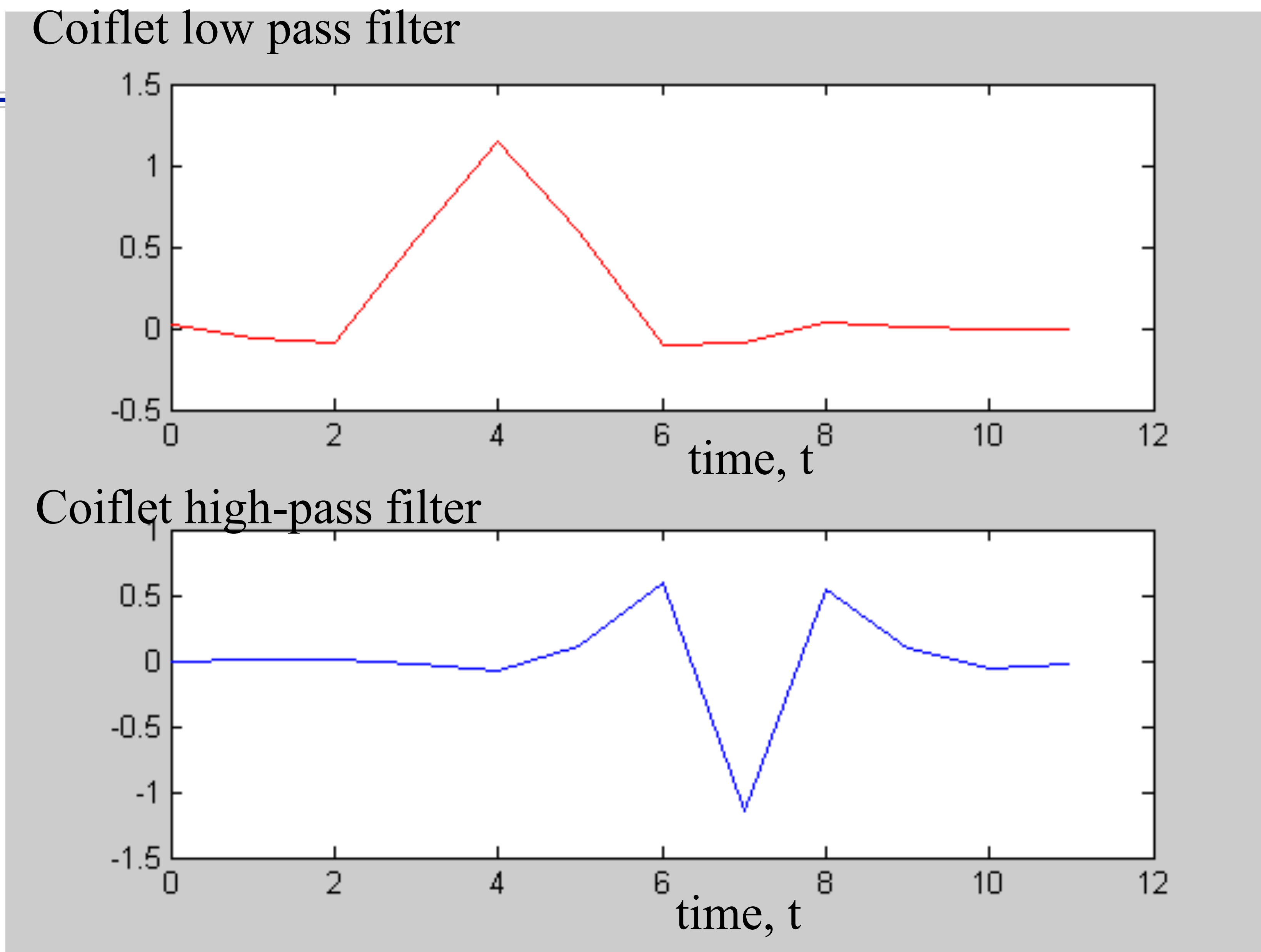


$\gamma(s_1,t)$: N/2 coefficients

$\gamma(s_2,t)$: N/4 coefficients

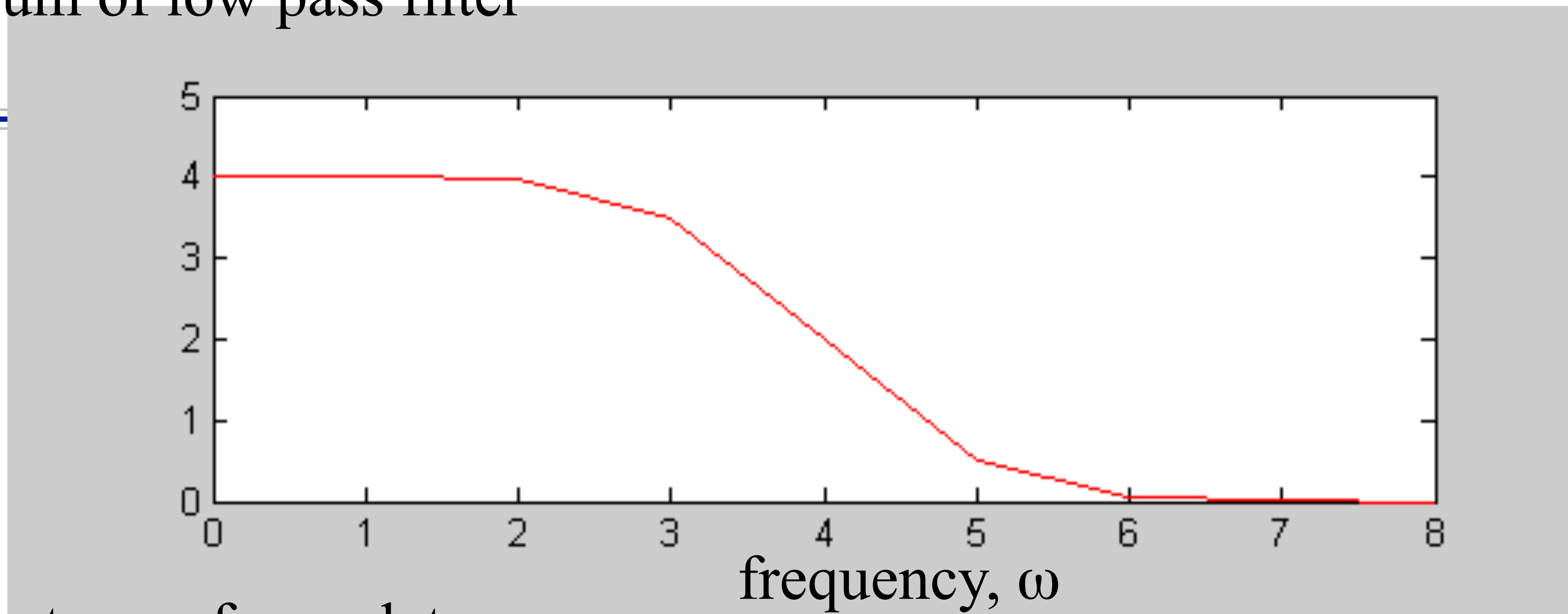
$\gamma(s_2,t)$: N/8 coefficients

Total: N coefficients

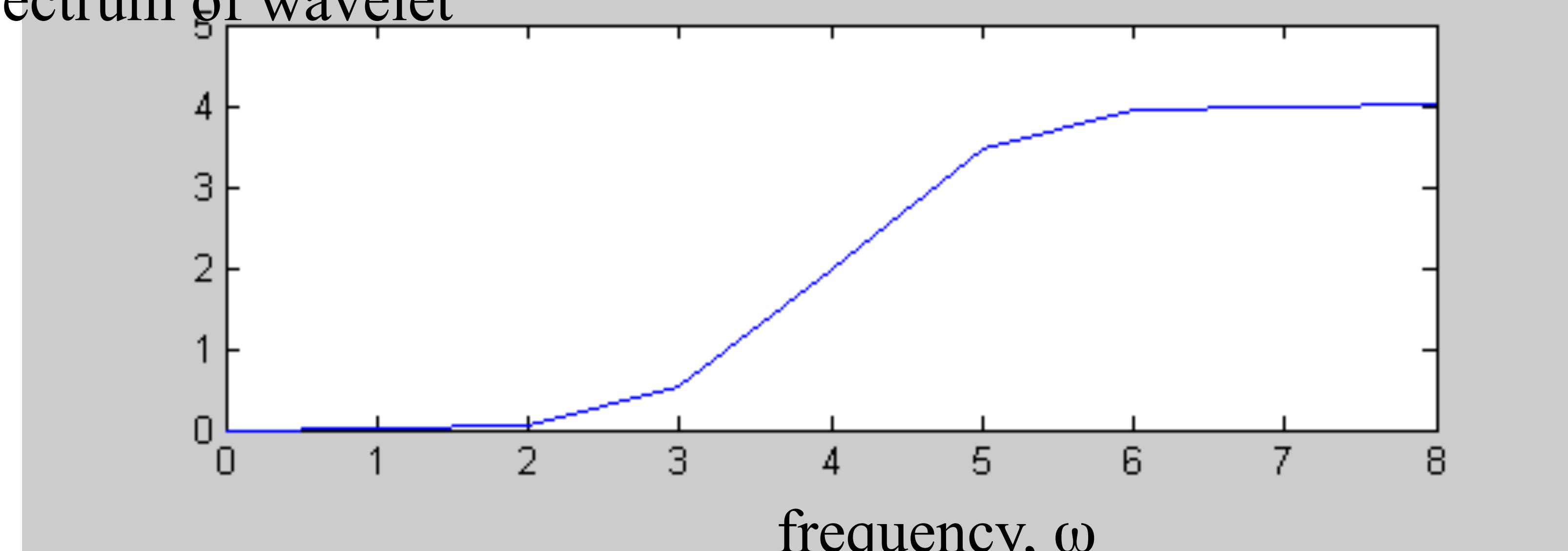


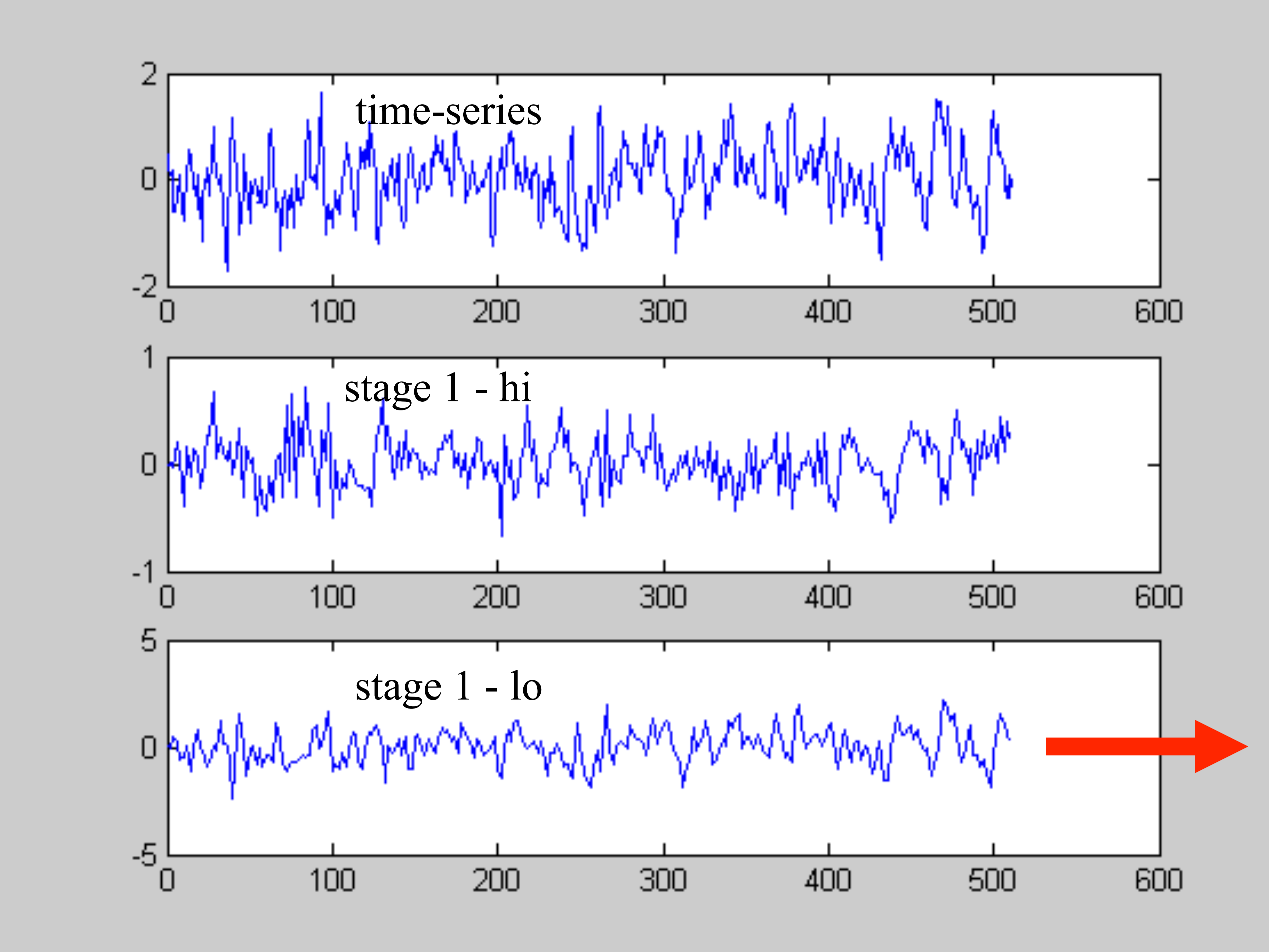
From http://en.wikipedia.org/wiki/Coiflet

Spectrum of low pass filter

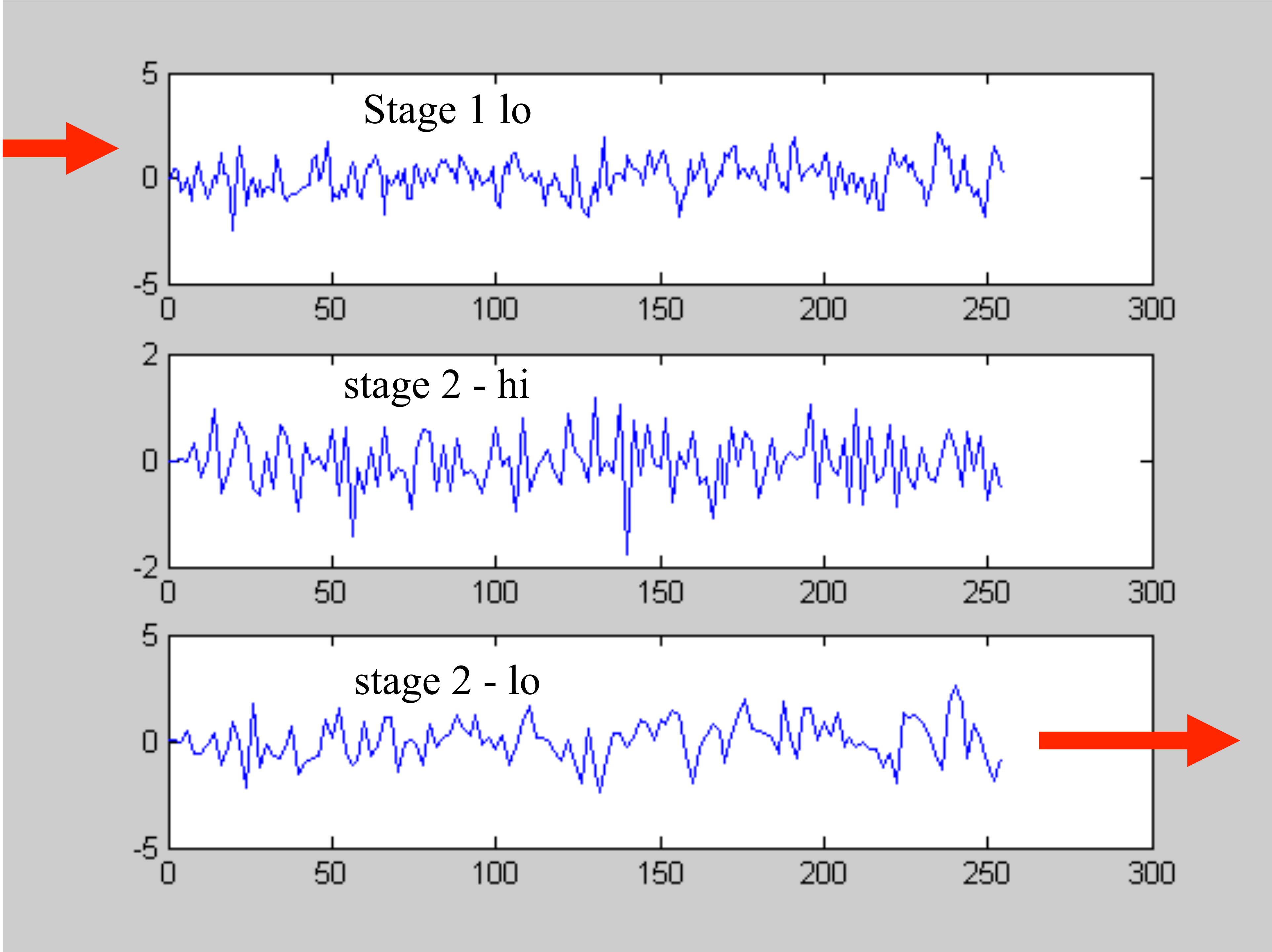


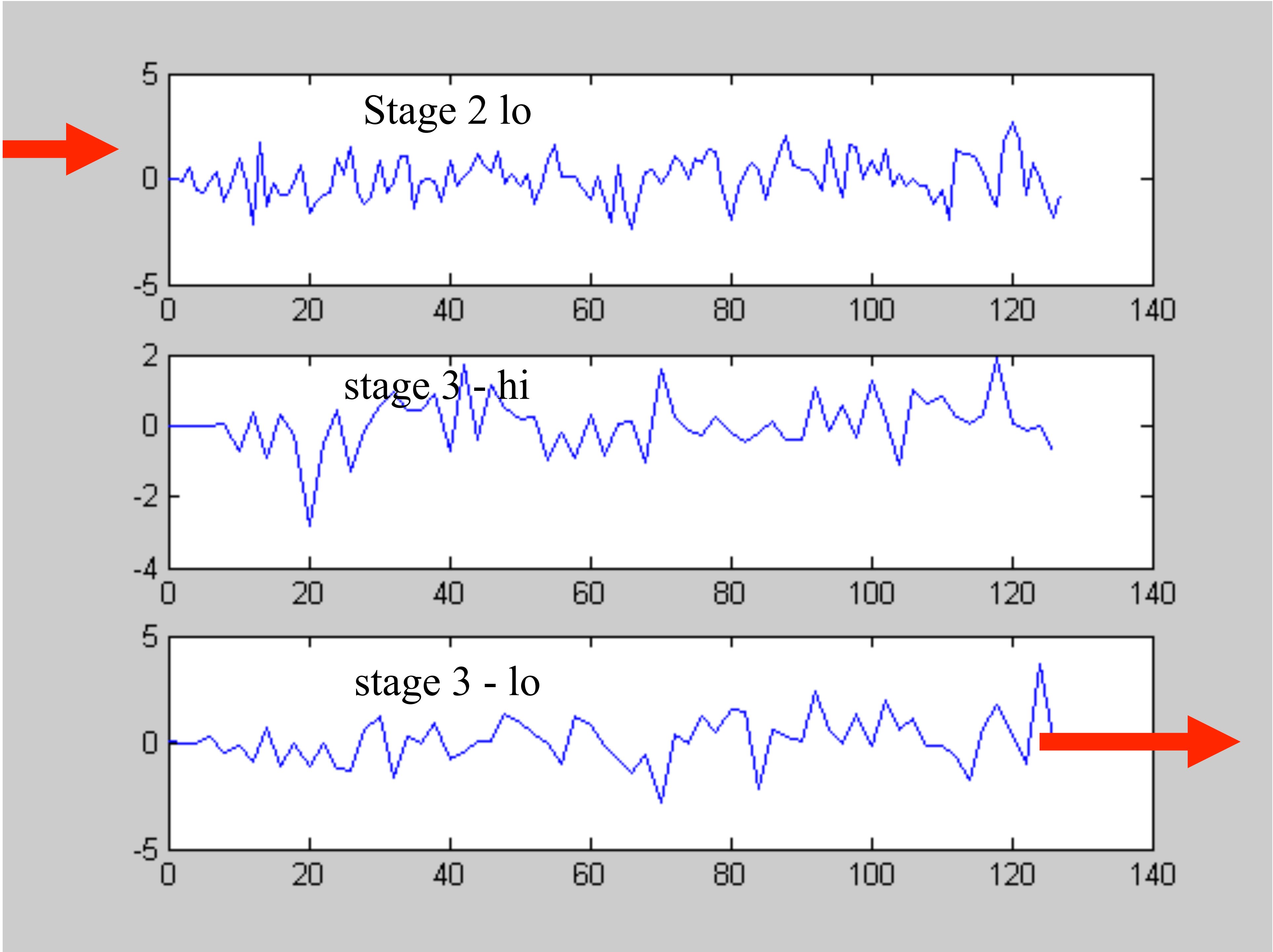
Spectrum of wavelet



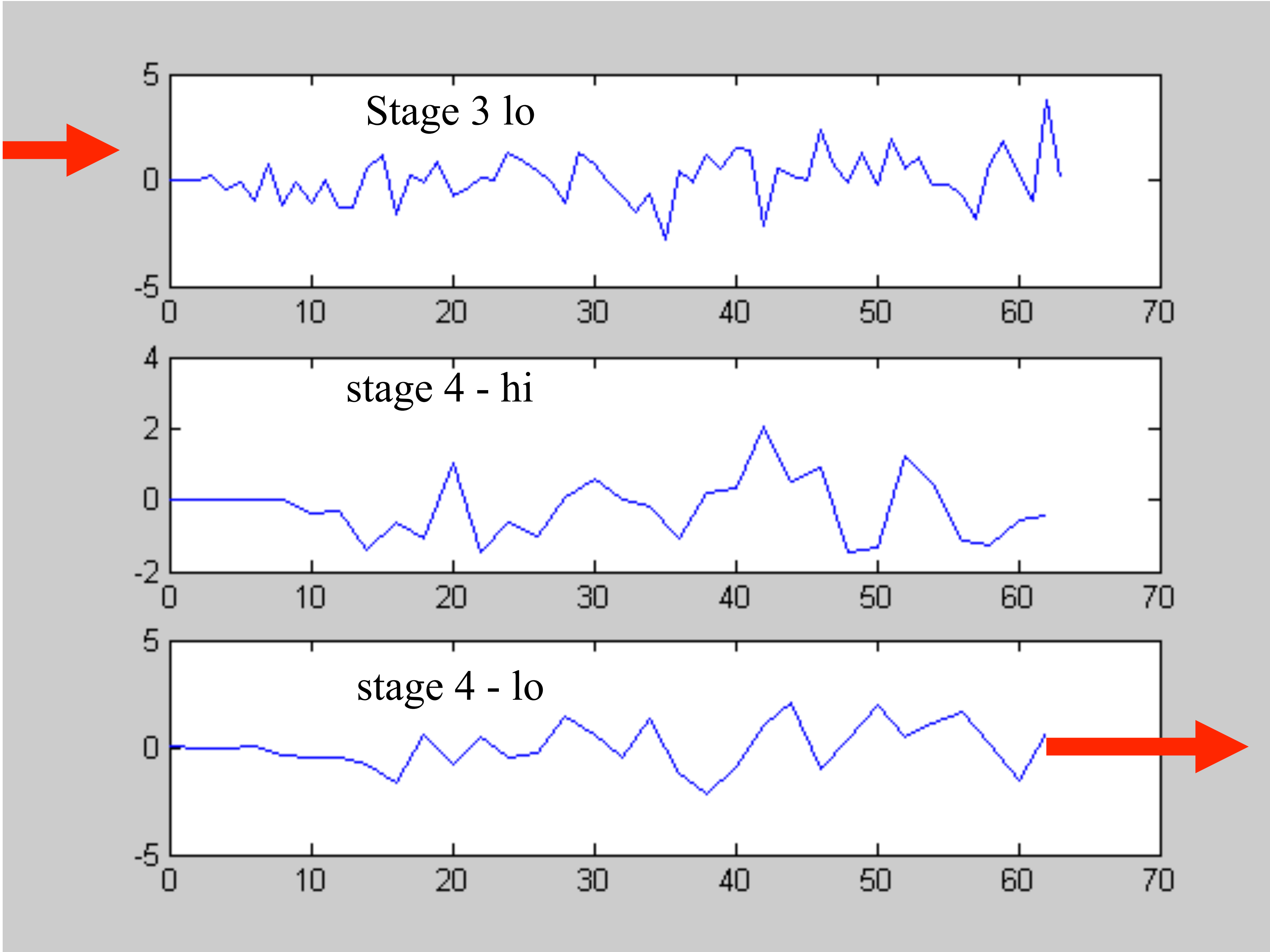


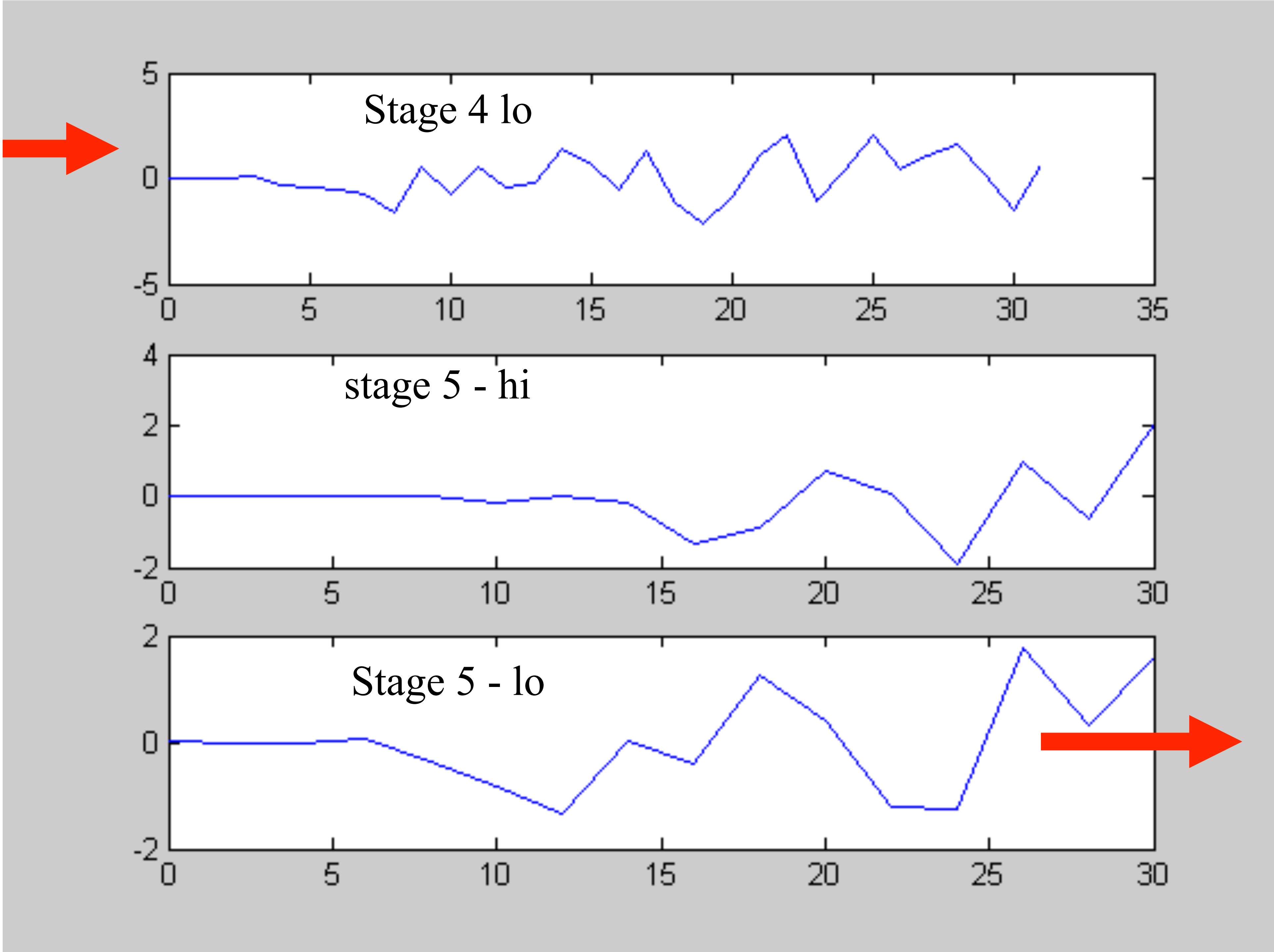
time-series
2
0
-2
0
100
200
300
400
500
600
stage 1 - hi
1
0
-1
0
100
200
300
400
500
600
stage 1 - lo
5
0
-5
0
100
200
300
400
500
600

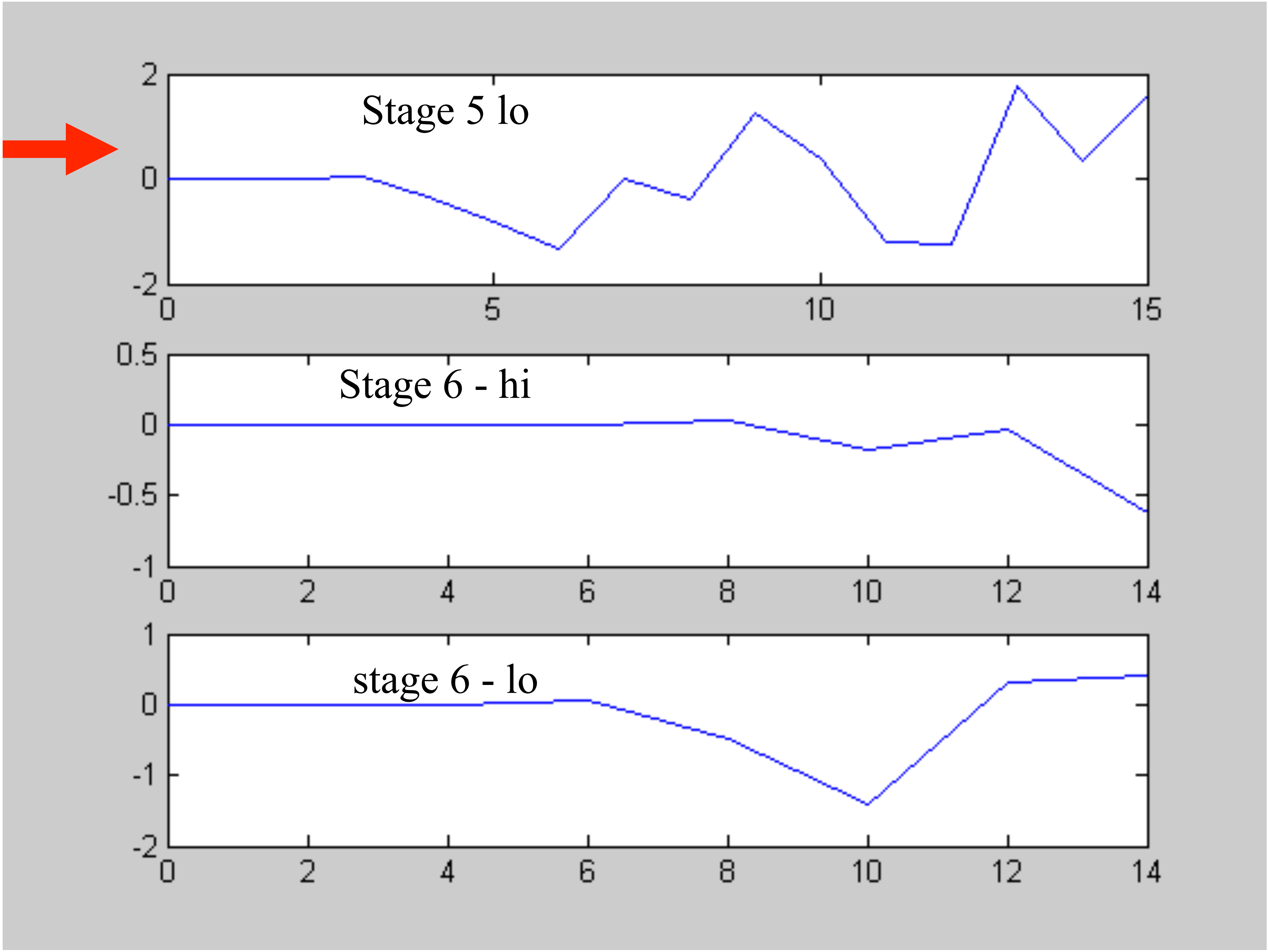

Stage 1 lo
stage 2 - hi
stage 2 - lo



Stage 2 lo
stage 3 - hi
stage 3 - lo
5
0
-5
2
-2
-4
0
20
40
60
80
100
120
140

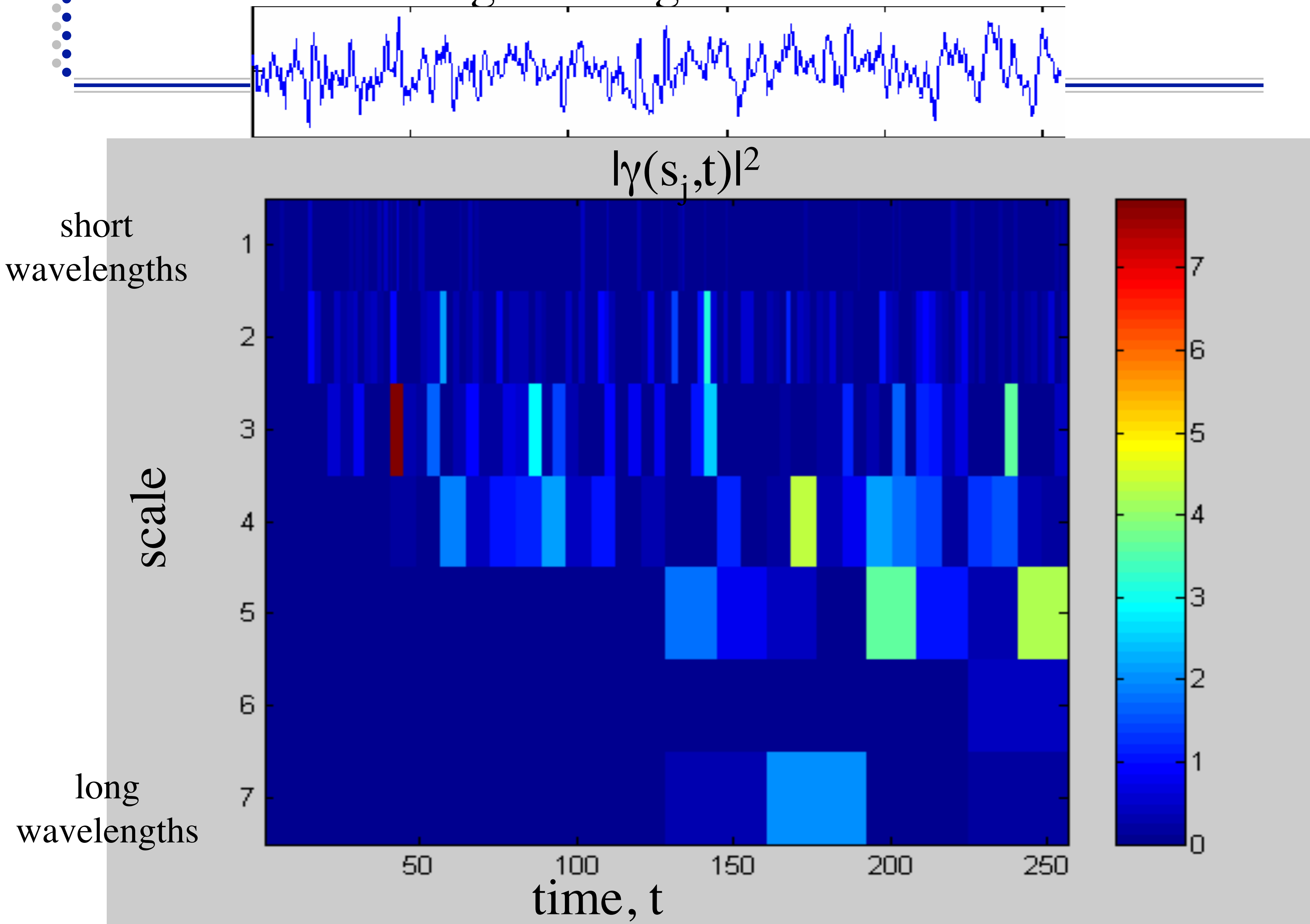

Stage 3 lo
stage 4 - hi
stage 4 - lo
5
0
-5
4
2
0
-2
0
10
20
30
40
50
60
70


Stage 4 lo
stage 5 - hi
Stage 5 - lo
5
0
-5
0
5
10
15
20
25
30
35
4
2
0
-2
0
5
10
15
20
25
30
2
0
-2
0
5
10
15
20
25
30



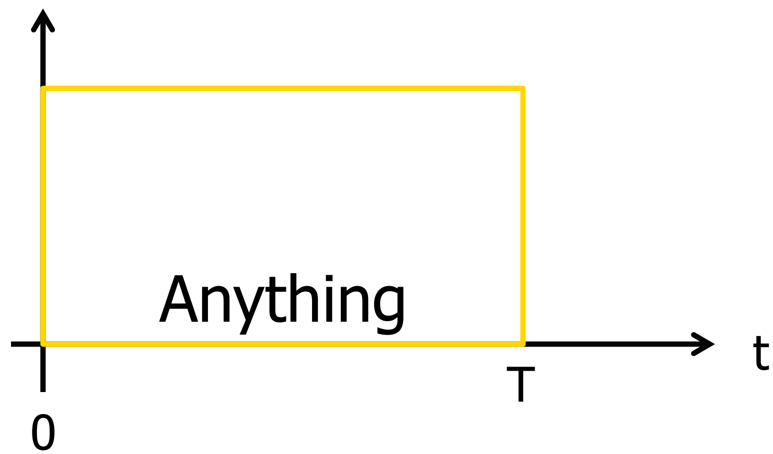
Stage 5 lo
Stage 6 - hi
stage 6 - lo

Putting it all together …



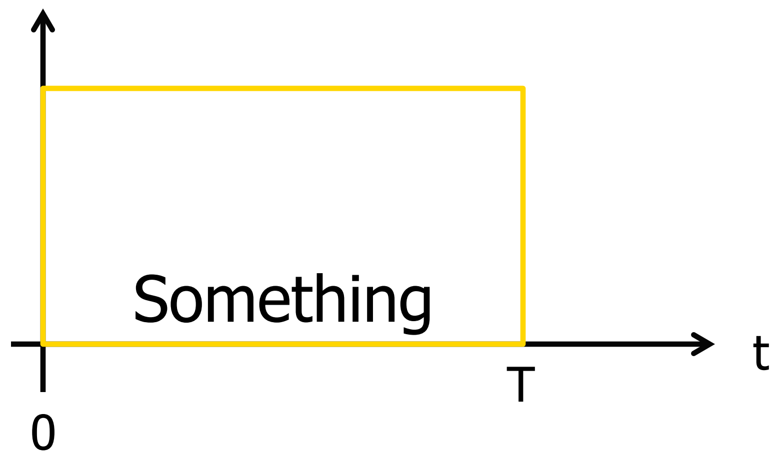
Compressive Sampling

Compressive Sampling



- What is the rate you need to sample at?
  - At least Nyquist

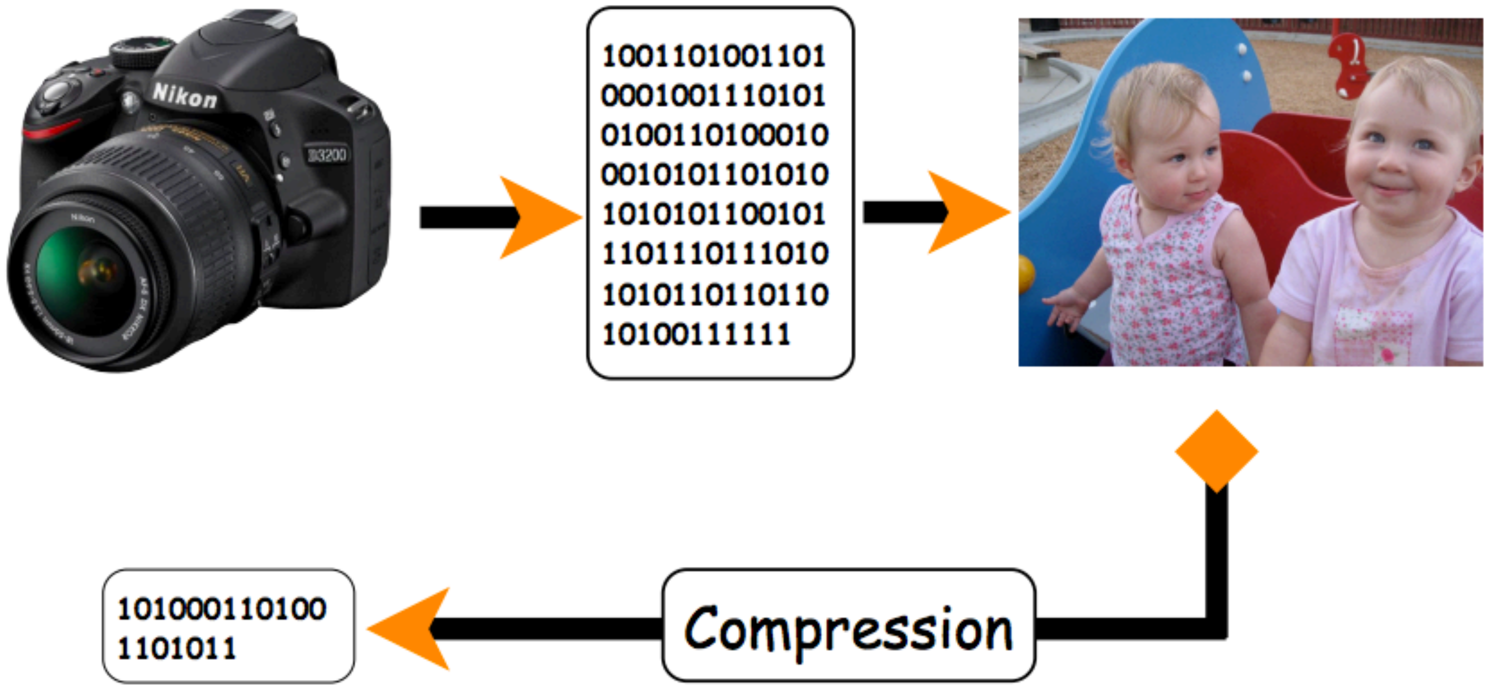
Compressive Sampling



- What is the rate you need to sample at?
  - Maybe less than Nyquist…

First: Compression

- Standard approach
  - First collect, then compress
    - Throw away unnecessary data



First: Compression

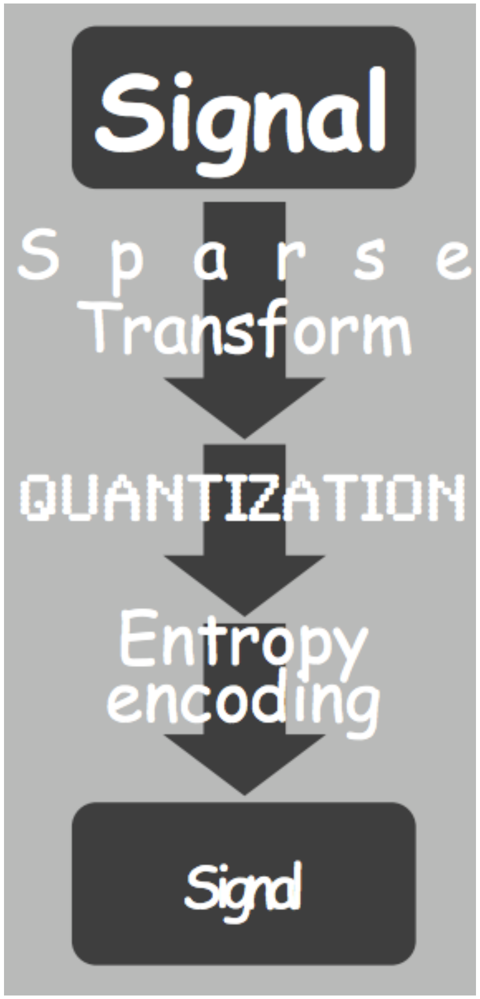
- Examples
  - Audio – 10x
    - Raw audio: 44.1kHz, 16bit, stereo = 1378 Kbit/sec
    - MP3: 44.1kHz, 16 bit, stereo = 128 Kbit/sec
  - Images – 22x
    - Raw image (RGB): 24bit/pixel
    - JPEG: 1280x960, normal = 1.09bit/pixel
  - Videos – 75x
    - Raw Video: (480x360)p/frame x 24b/p x 24frames/s + 44.1kHz x 16b x 2 = 98,578 Kbit/s
    - MPEG4: 1300 Kbit/s

First: Compression

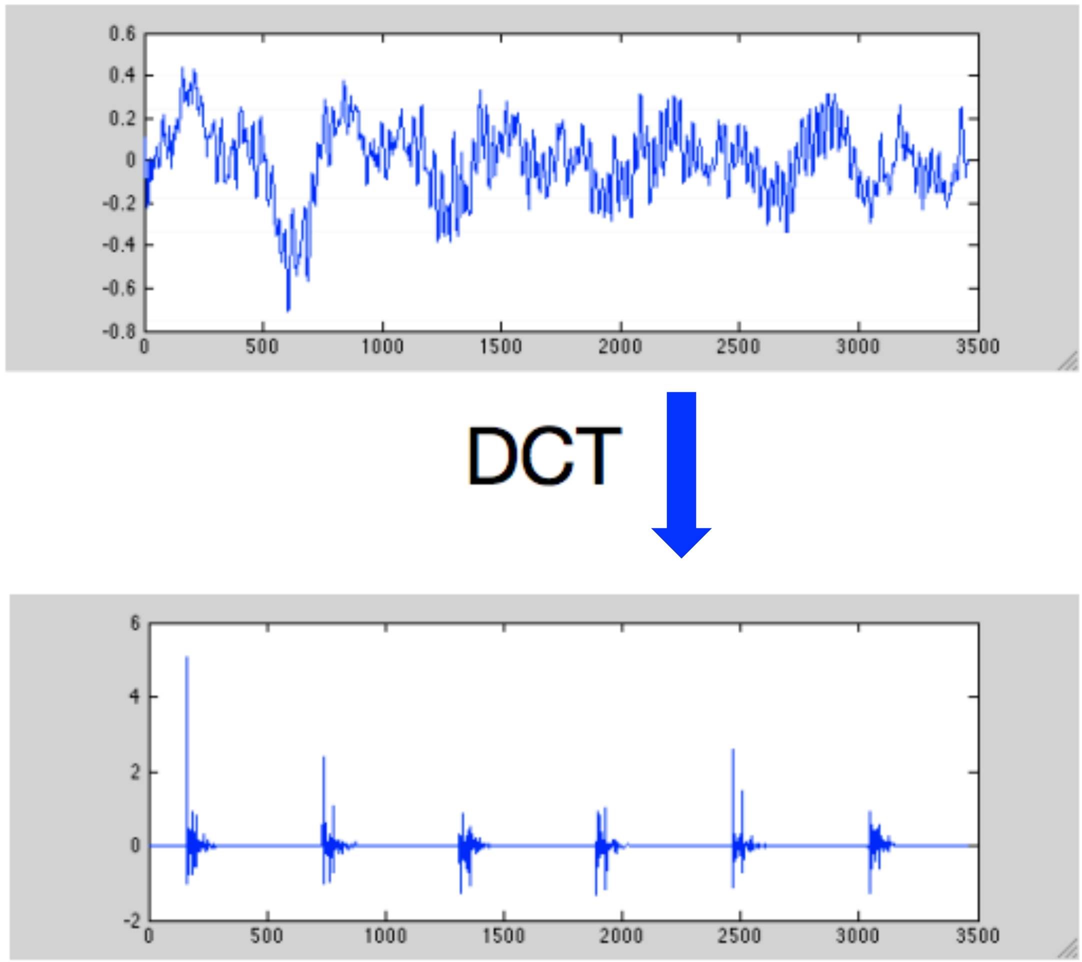
- Almost all compression algorithm use transform coding
  - mp3: DCT
  - JPEG: DCT
  - JPEG2000: Wavelet
  - MPEG: DCT & time-difference

First: Compression

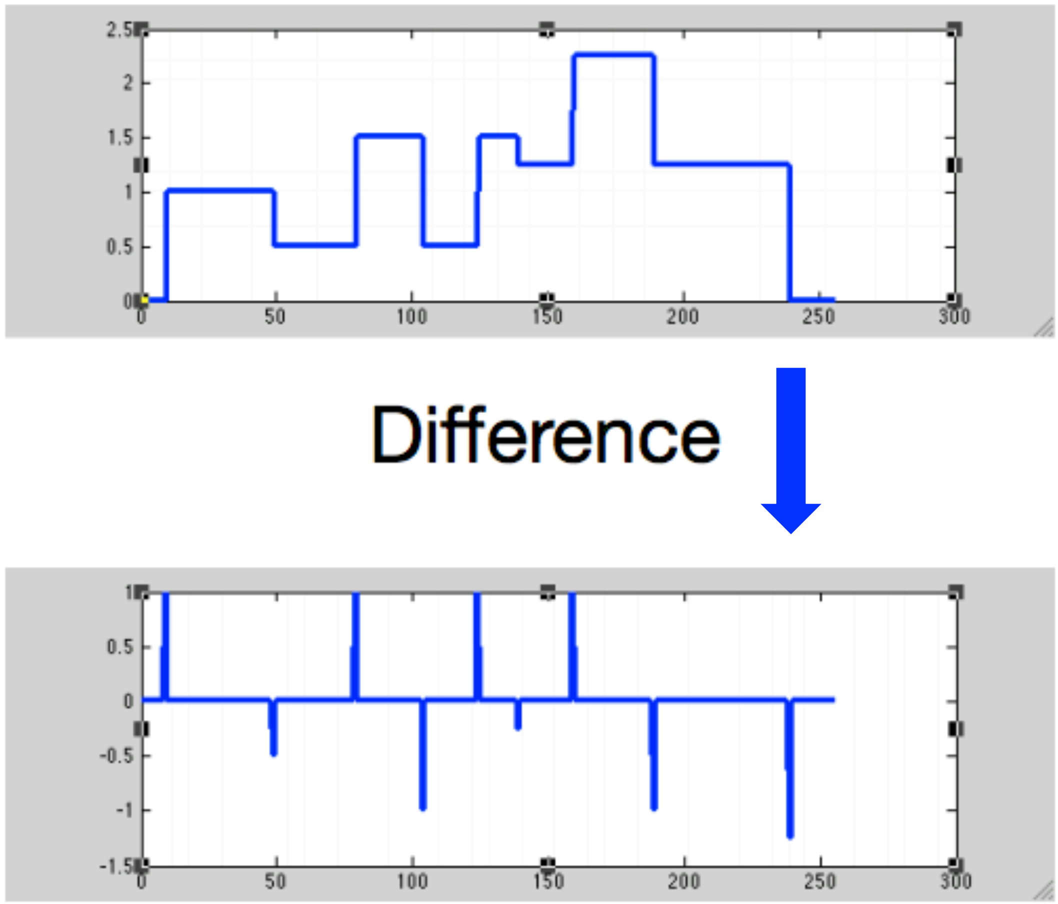
- Almost all compression algorithm use transform coding
  - mp3: DCT
  - JPEG: DCT
  - JPEG2000: Wavelet
  - MPEG: DCT & time-difference



Sparse Transform

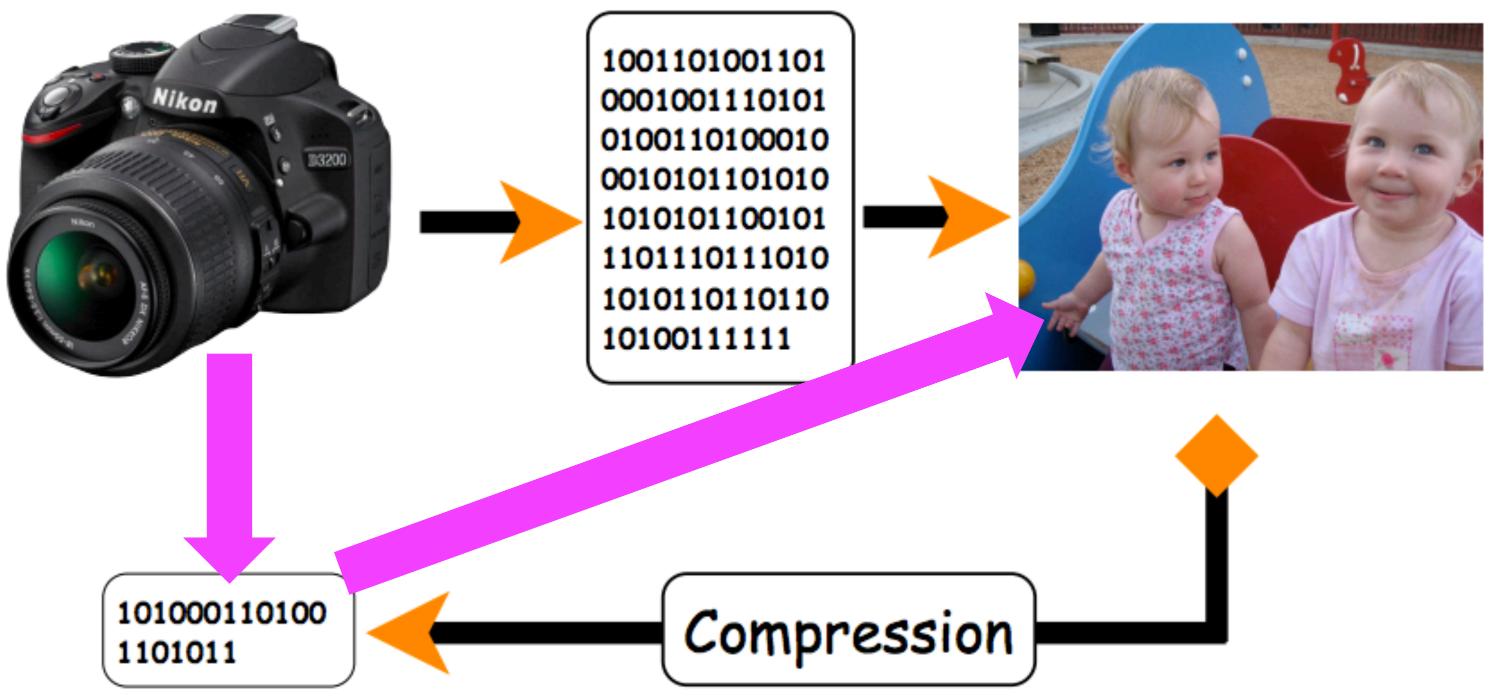


Sparse Transform



Compressive Sensing/Sampling

- Standard approach
  - First collect, then compress
    - Throw away unnecessary data



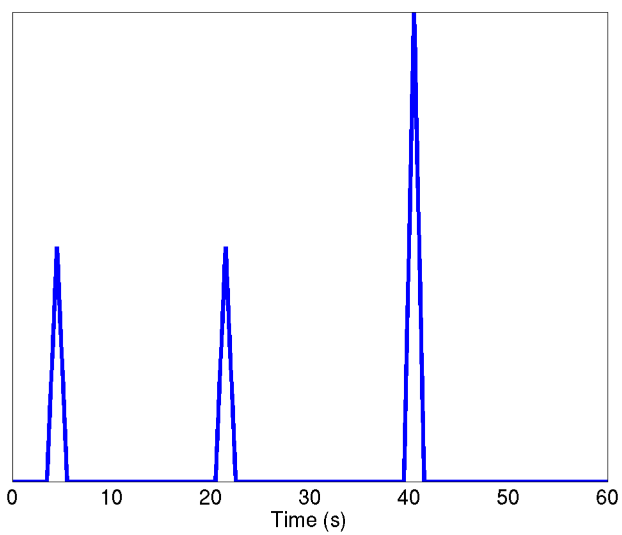
Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in some cases exactly recover

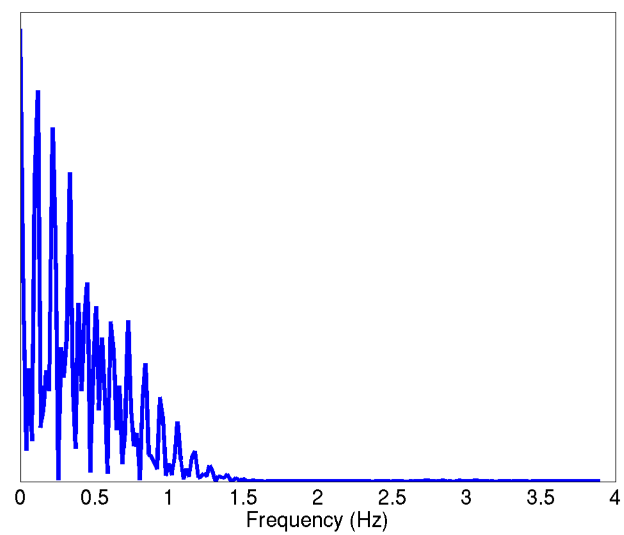
Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in some cases exactly recover

Sparse signal in time



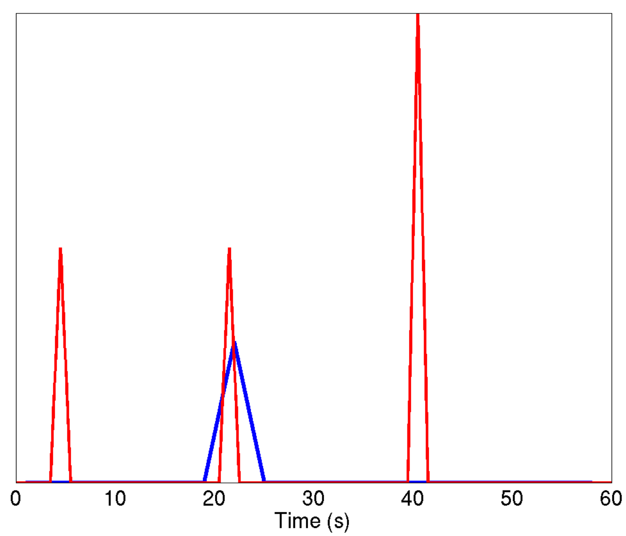
Frequency spectrum



Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in some cases exactly recover

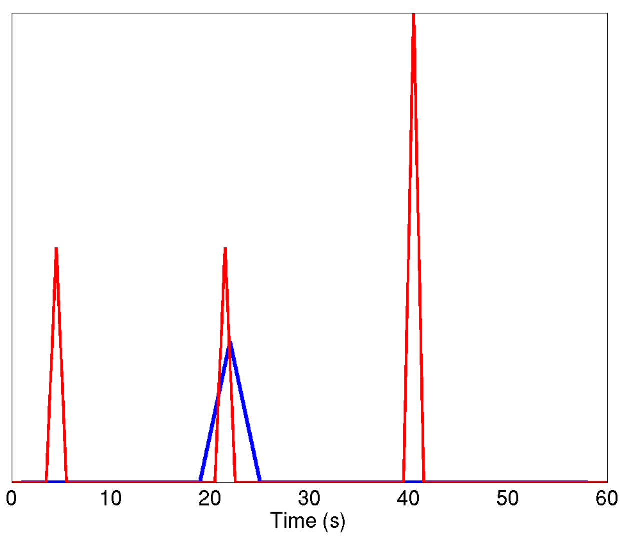
Undersampled in time

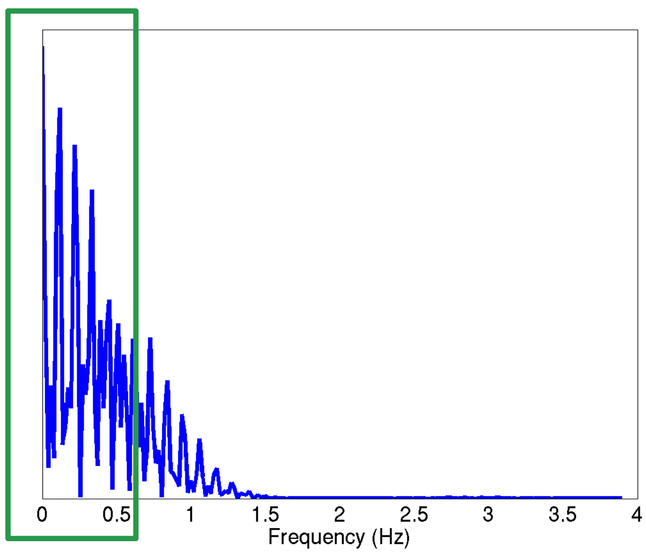


Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in some cases exactly recover

Undersampled in time

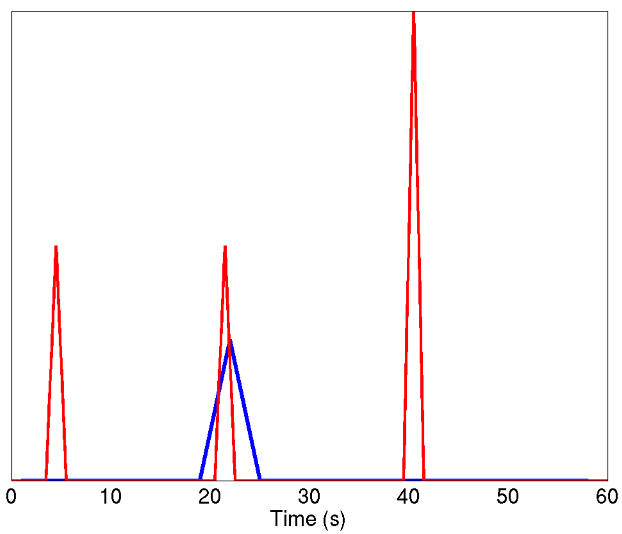




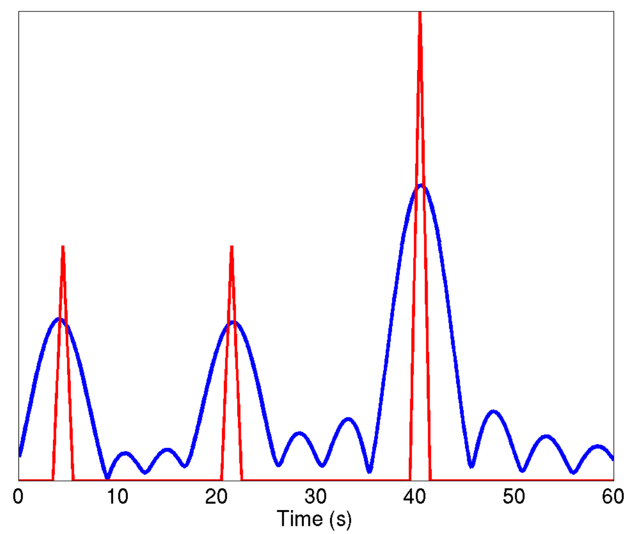
Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in some cases exactly recover

Undersampled in time



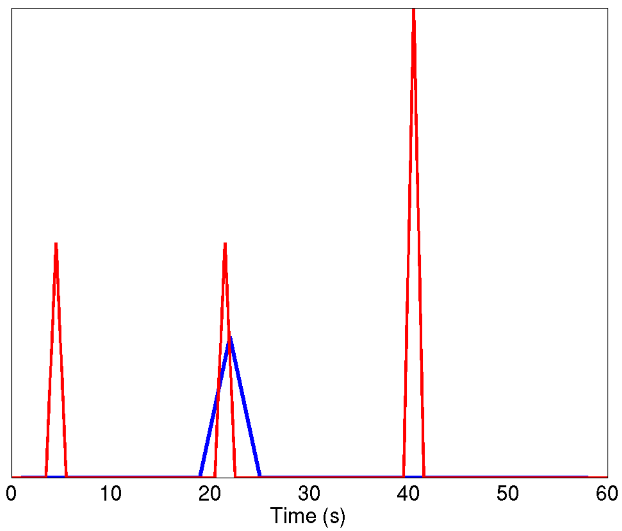
Undersampled in frequency
(reconstructed in time with IFFT)



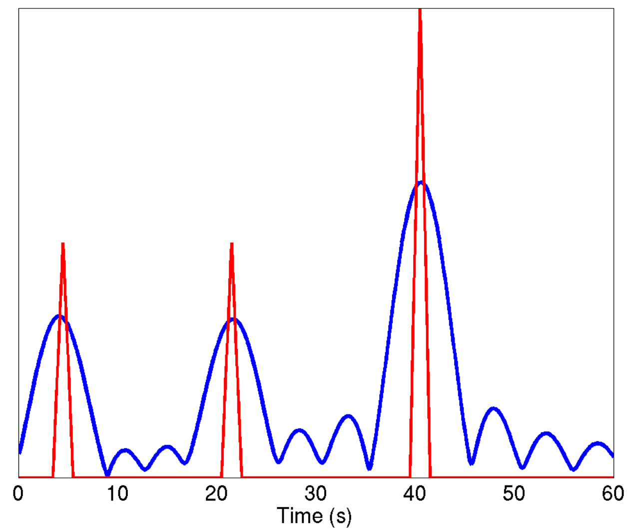
Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in some cases exactly recover

Undersampled in time

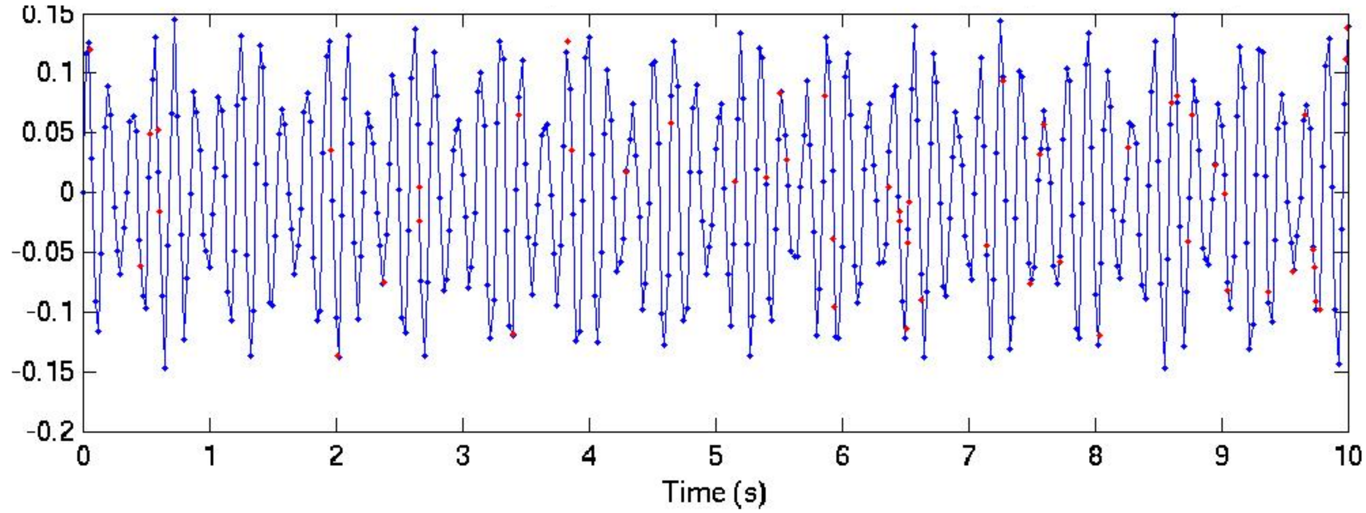


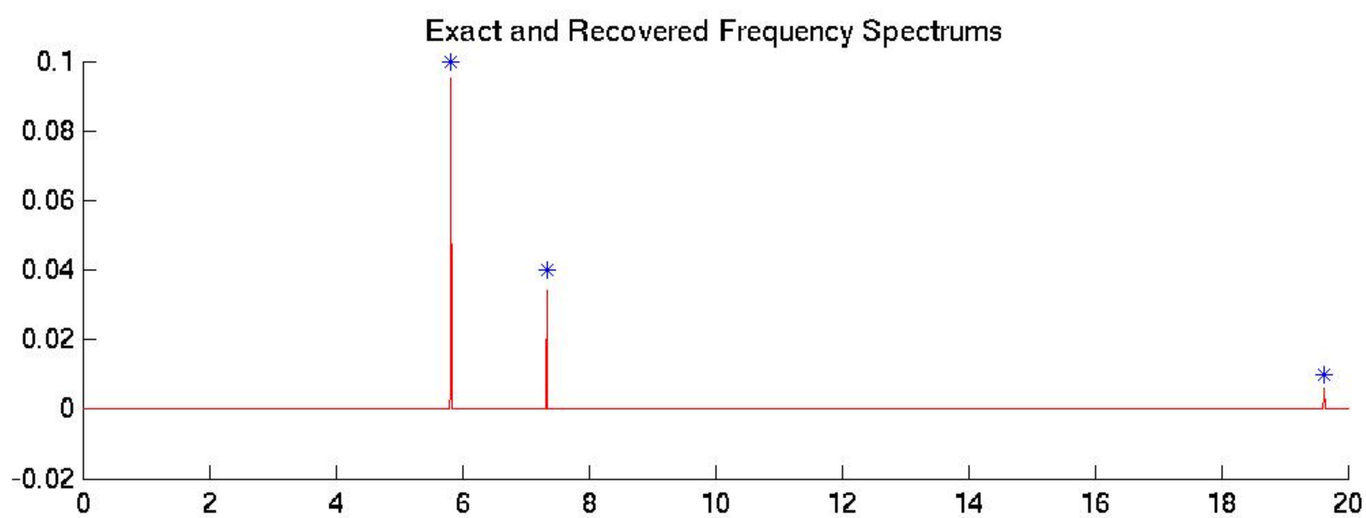
Undersampled in frequency
(reconstructed in time with IFFT)



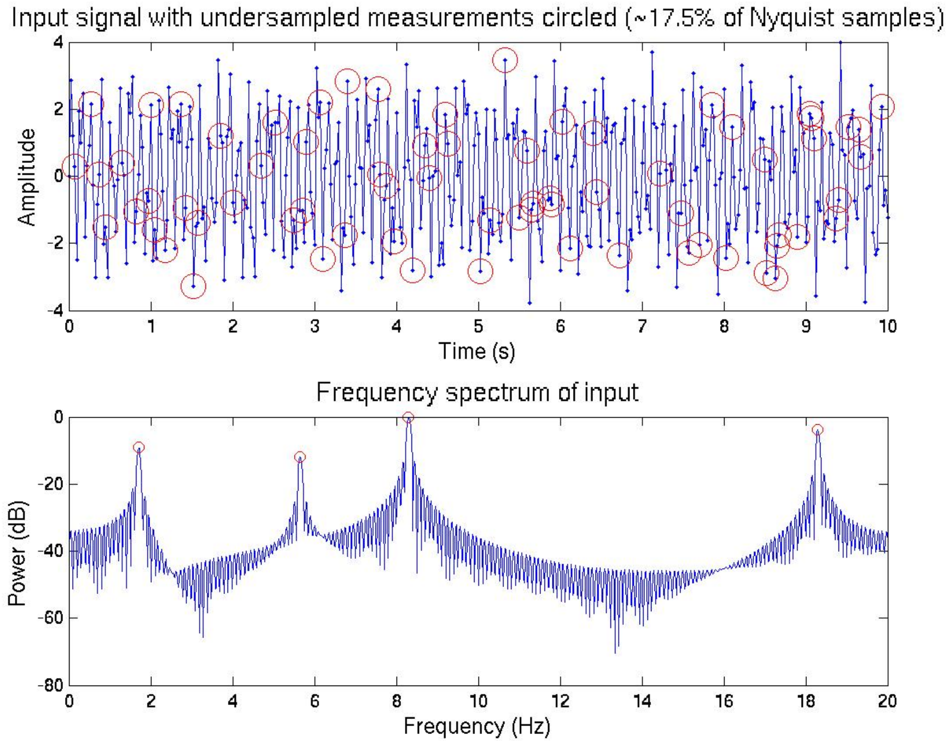
Requires sparsity and incoherent sampling

Compressive Sampling: Simple Example





Compressive Sampling

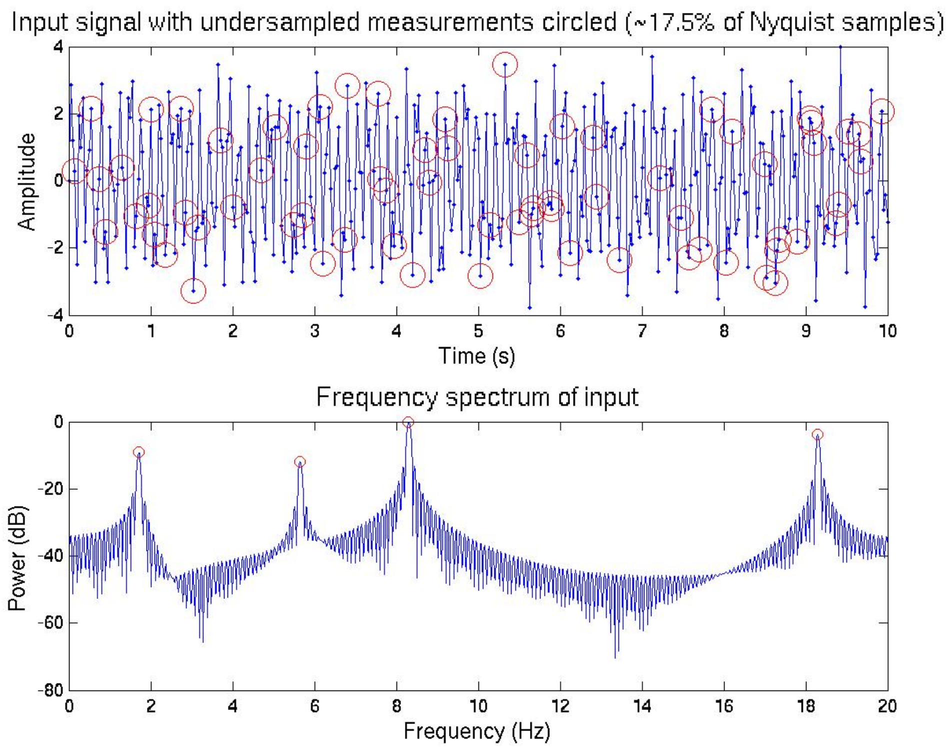


- Sense signal randomly M times
  - $M > C \cdot \mu^2(\Phi, \Psi) \cdot S \cdot \log N$
- Recover with linear program

$$\min \sum_{\omega} |\hat{g}(\omega)| \quad \text{subject to} \quad g(t_m) = f(t_m), \quad m = 1, \ldots, M$$

Compressive Sampling

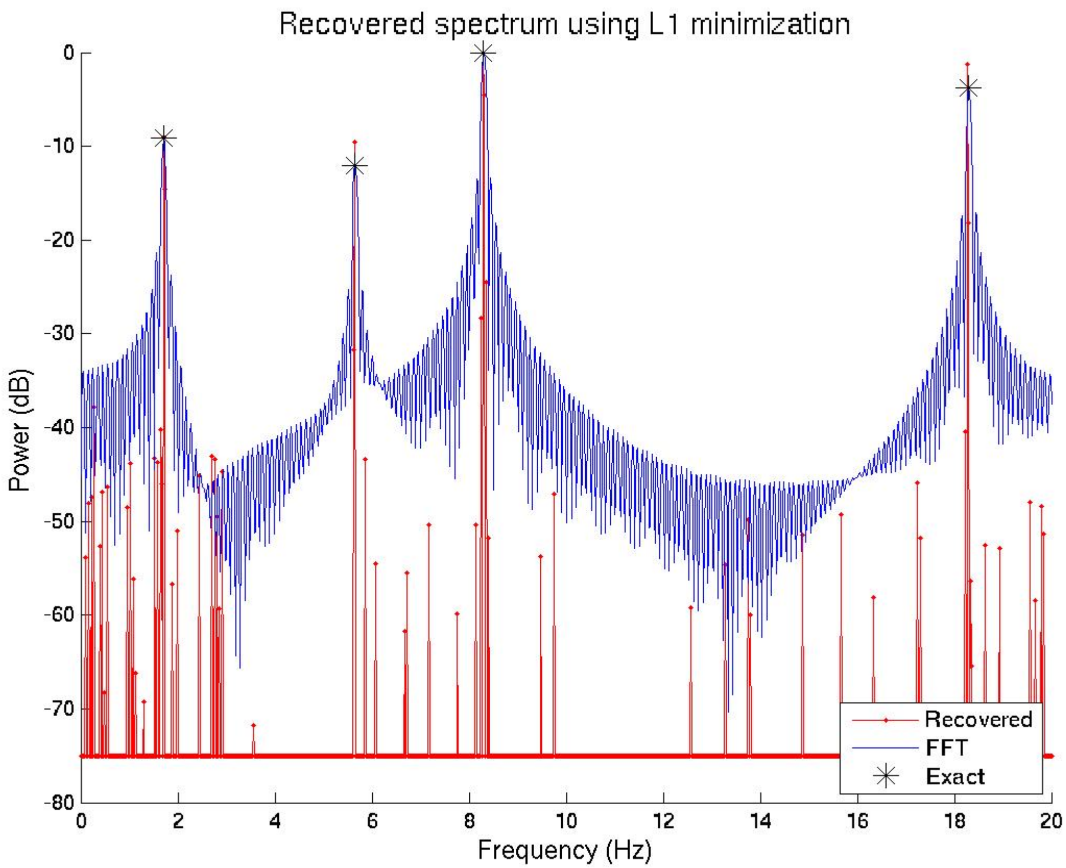
$$\hat{f}(\omega) = \sum_{i=1}^{K} \alpha_i \delta(\omega_i - \omega) \overset{\mathcal{F}}{\Leftrightarrow} f(t) = \sum_{i=1}^{K} \alpha_i e^{i\omega_i t}$$



- Sense signal randomly M times
  - $M > C \cdot \mu^2(\Phi, \Psi) \cdot S \cdot \log N$
- Recover with linear program

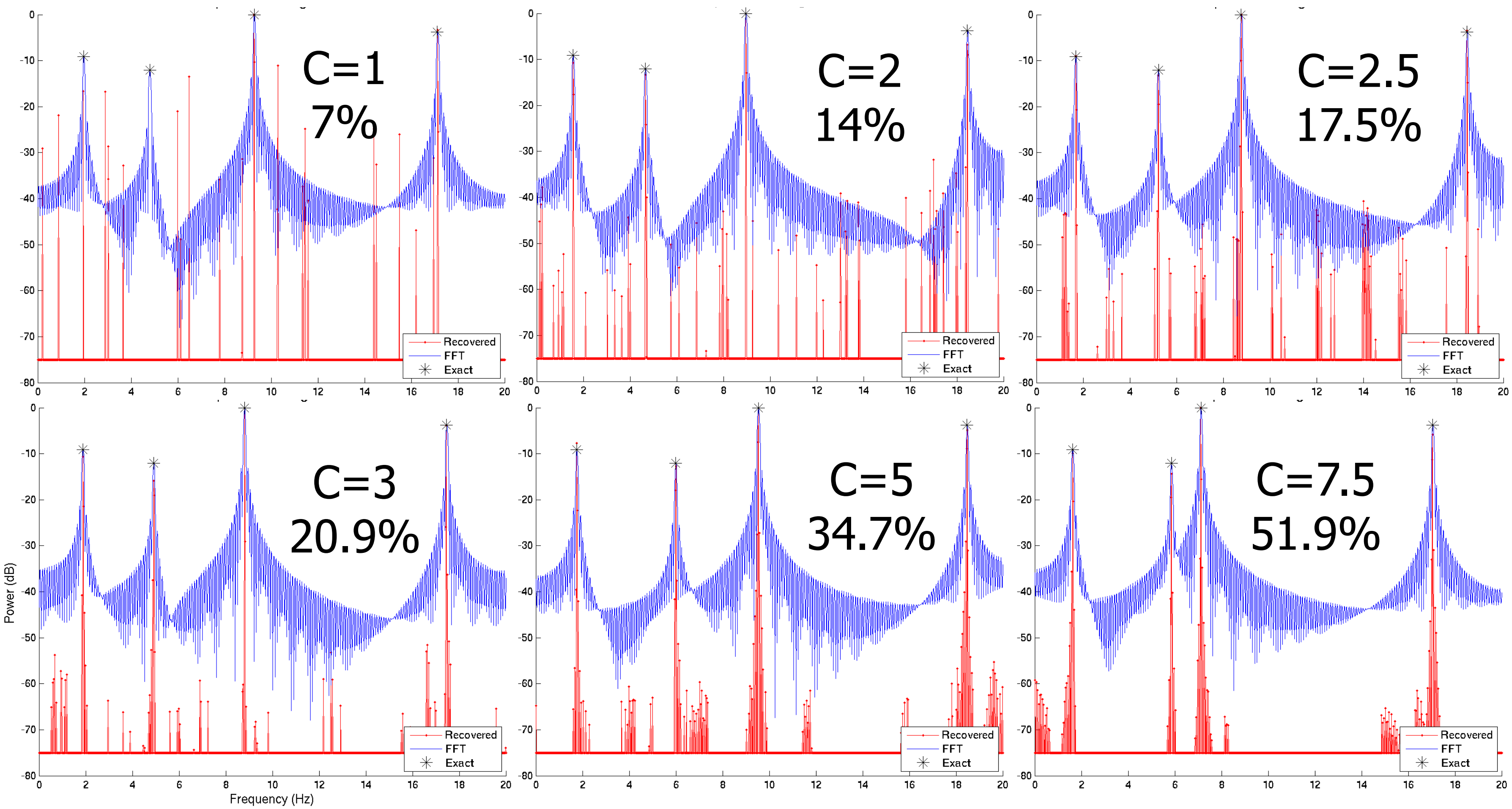
$$\min \sum_{\omega} |\hat{g}(\omega)| \quad \text{subject to} \quad g(t_m) = f(t_m), \quad m = 1, \ldots, M$$

Example: Sum of Sinusoids

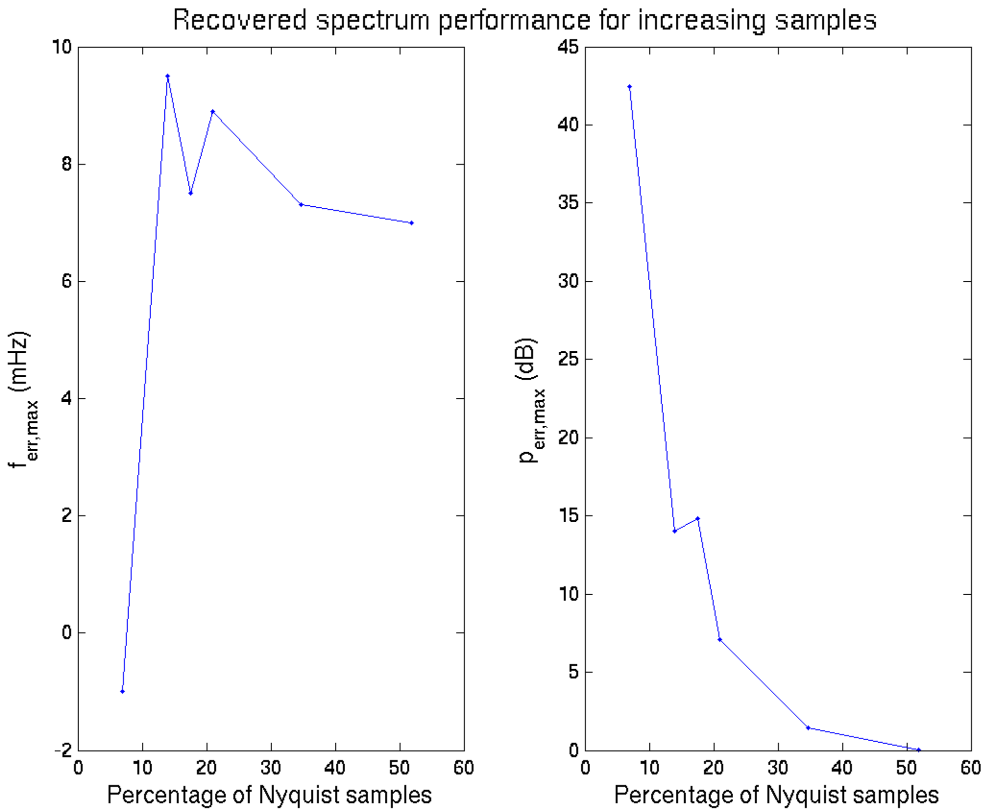


- Two relevant "knobs"
  - percentage of Nyquist samples as altered by adjusting optimization factor, *C*
  - input signal duration, *T*
    - Data block size

Example: Increasing C

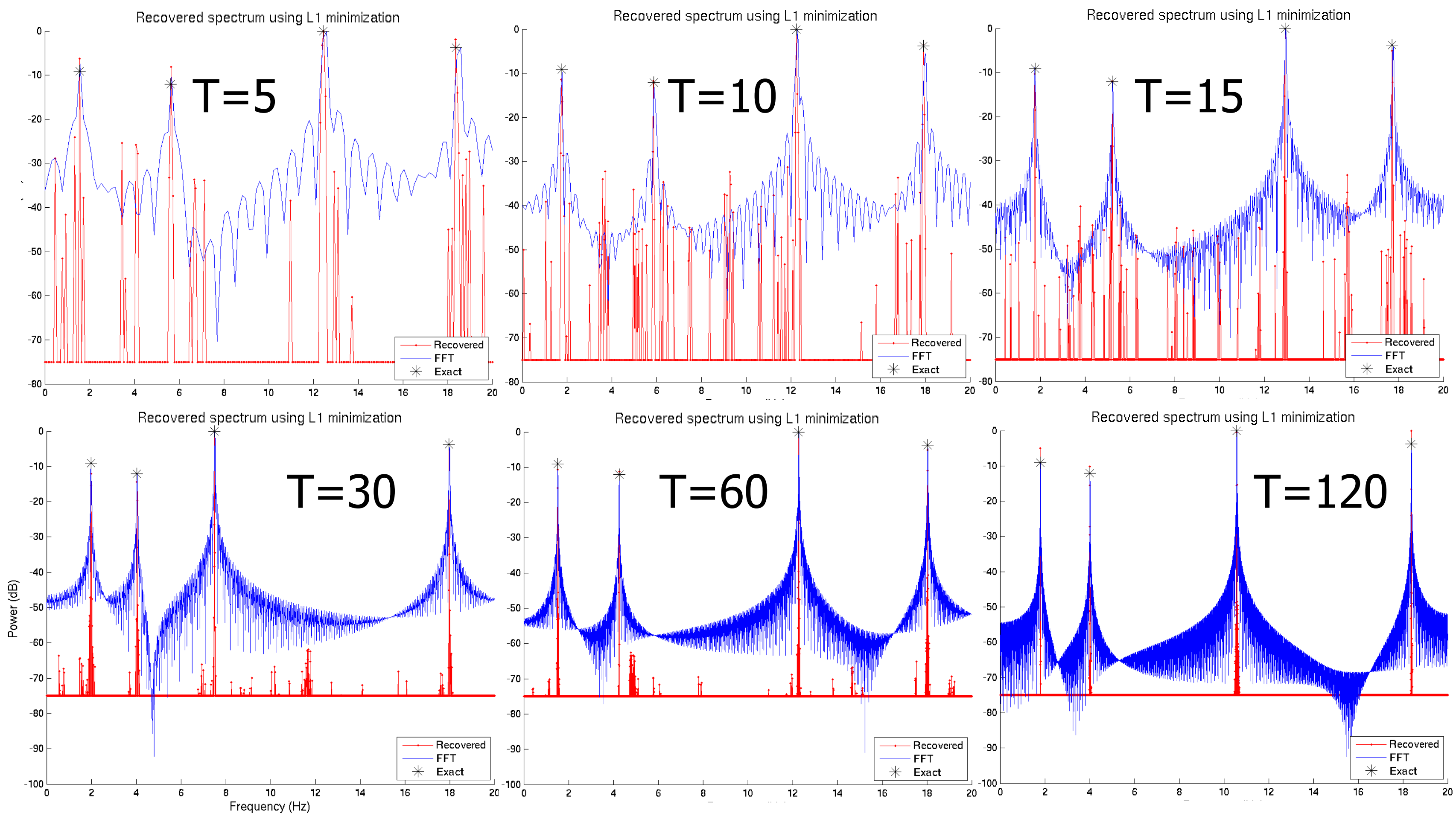


Example: Increasing C

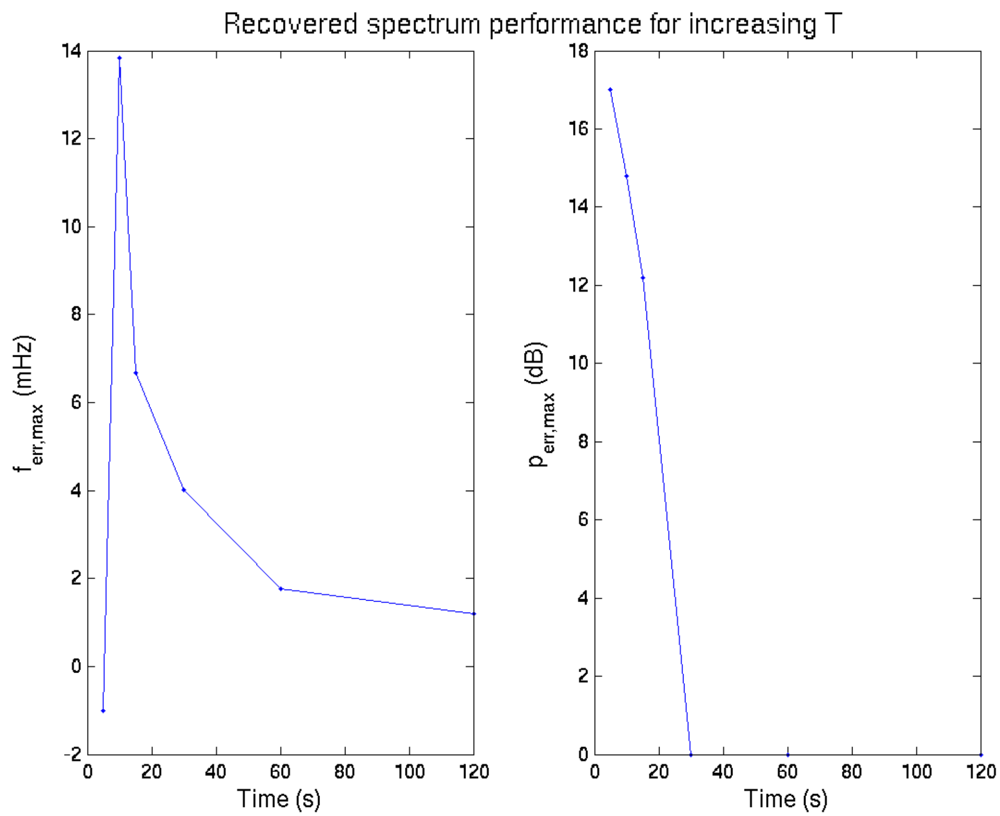


- **$f_{err,max}$ within 10 mHz**
- **$p_{err,max}$ decreasing**

Example: Increasing T

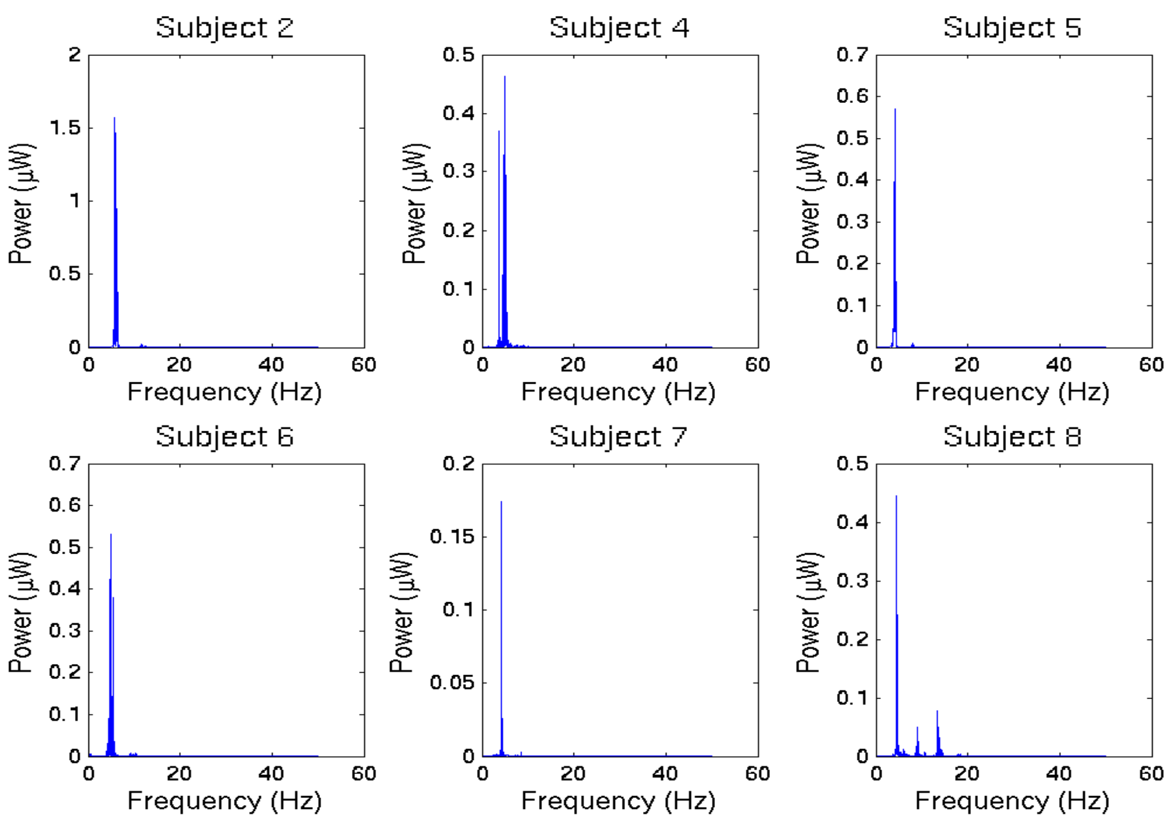


Example: Increasing T



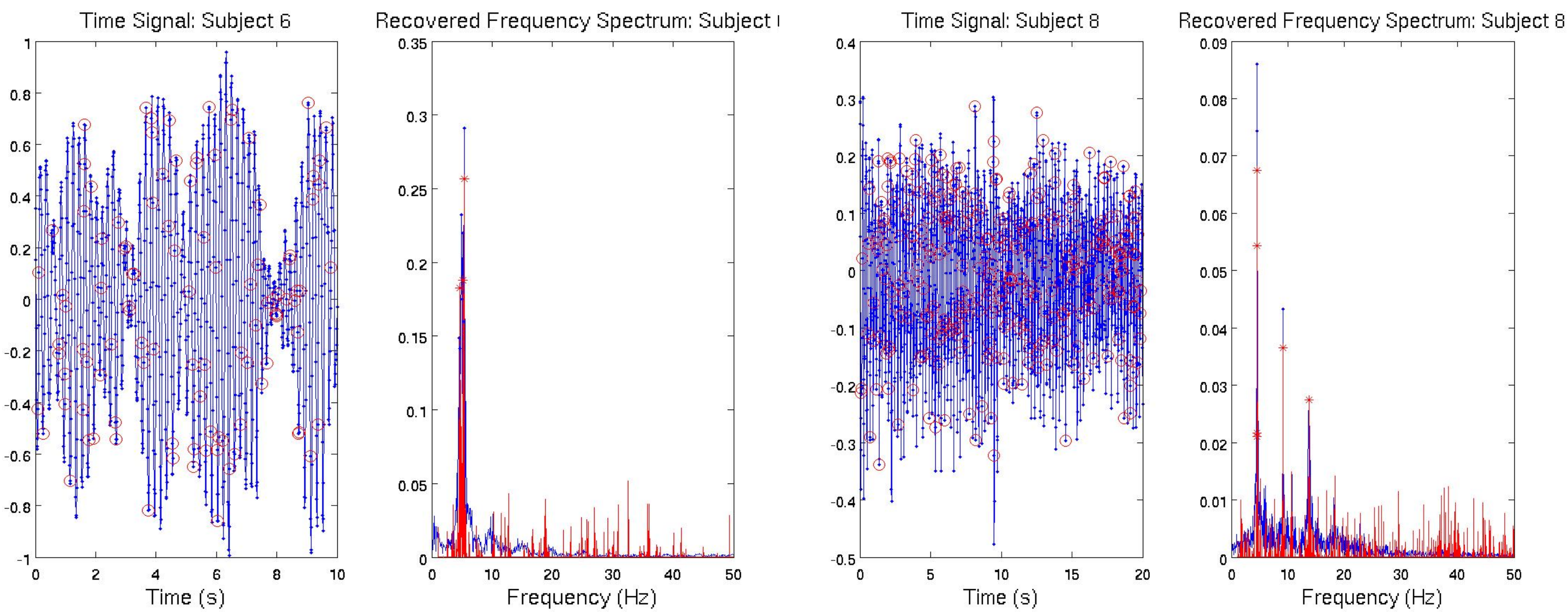
- $f_{err,max}$ **decreasing**
- $p_{err,max}$ **decreasing**

Biometric Example: Parkinson's Tremors

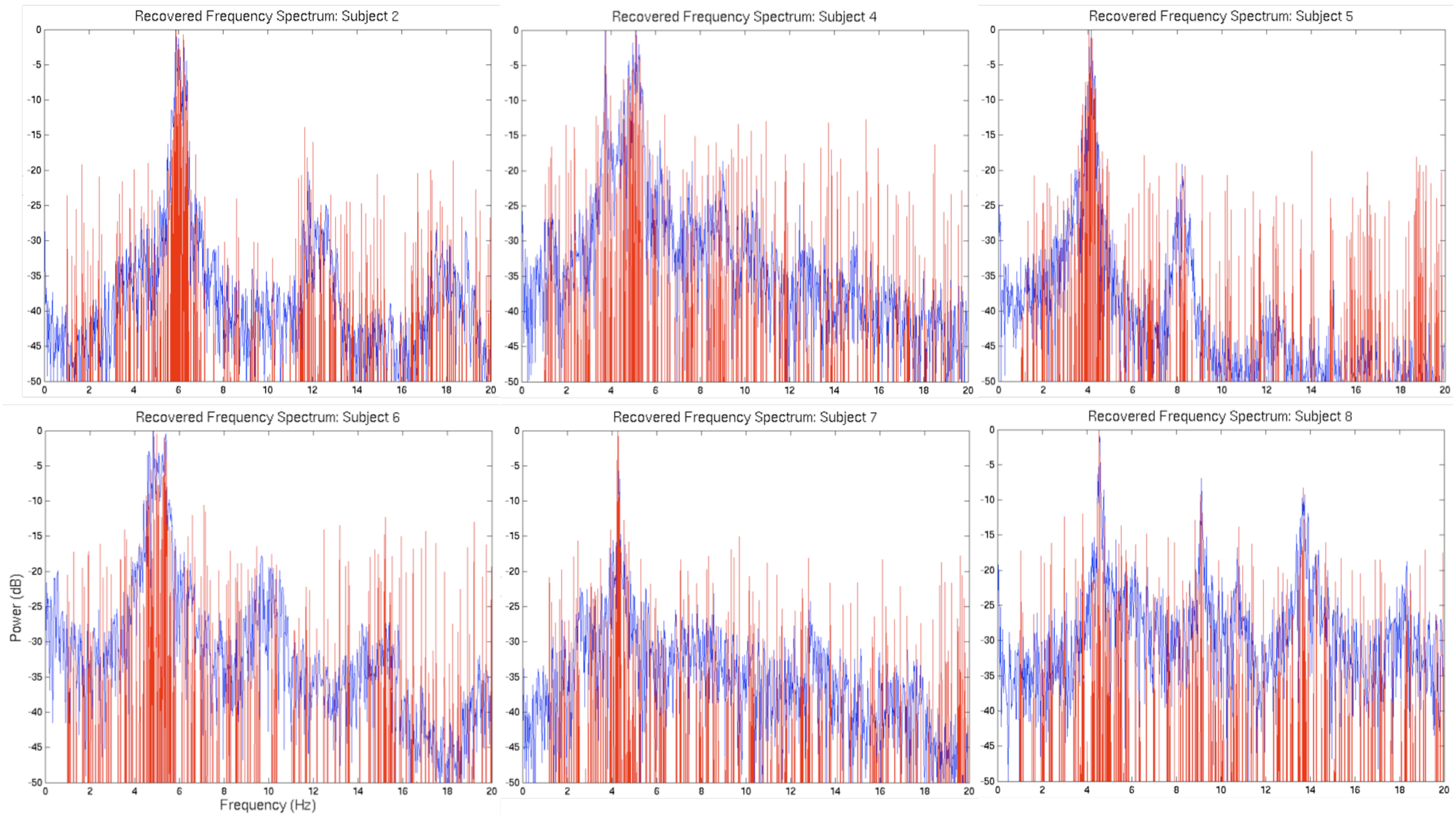


- 6 Subjects of real tremor data
  - collected using low intensity velocity-transducing laser recording aimed at reflective tape attached to the subjects' finger recording the finger velocity
  - All show Parkinson's tremor in the 4-6 Hz range.
  - Subject 8 shows activity at two higher frequencies
  - Subject 4 appears to have two tremors very close to each other in frequency

Compressive Sampling: Real Data

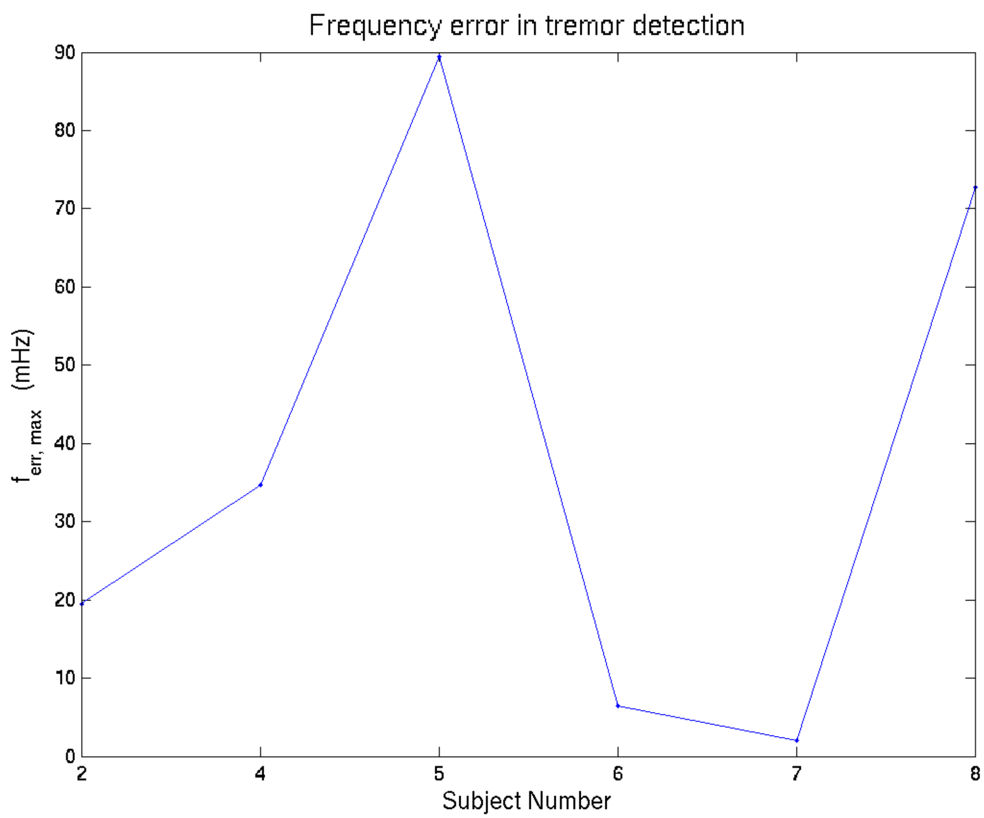


Biometric Example: Parkinson's Tremors



- **C=10.5, T=30**
  - 20% Nyquist required samples

Biometric Example: Parkinson's Tremors



- **Tremors detected within 100 mHz**
- **randomly sample 20% of the Nyquist required samples**

Requires post processing to randomly sample!

Implementing Compressive Sampling

- Devised a way to randomly sample 20% of the Nyquist required samples and still detect the tremor frequencies within 100mHz
  - Requires post processing to randomly sample!

- Implement hardware on chip to "choose" samples in real time
  - Only write to memory the "chosen" samples
    - Design random-like sequence generator
  - Only convert the "chosen" samples
    - Design low energy ADC

Big Ideas

- Wavelet transform
  - Capture temporal data with fewer coefficients than STFT
- Compressive Sampling
  - Integrated sensing/sampling, compression and processing
  - Based on sparsity and incoherency

Admin

- Tania extra office hours
  - Monday 1-3pm
- Tuesday lecture
  - Review
  - Final exam details

- Project
  - Due 4/24